Master Thesis

EIT InnoEnergy Renewable Energy (RENE)

Design and implementation of a supervisory control and data acquisition system (SCADA) for a microgrid laboratory

Author: Benedek Pásztor
Supervisor: Alba Colet Subirachs
Director: Oriol Gomis Bellmunt
Call: June 2018

ETSEIB

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

MSc RENE is a cooperation between

Universitat Politècnica de Catalunya, Spain | KTH-Royal Institute of Technology, Sweden
Instituto Superior Técnico, Portugal | École Polytechnique (ParisTech), France

Abstract

This report presents the work conducted as a master thesis project within SmartLab laboratory of the Catalonian Institute for Energy Research (IREC). The focus of the work is on the design and development of a Supervisory Control and Data Acquisition (SCADA) system for the Institute's laboratory microgrid system. Based on the pre-defined requirements, a comprehensive system has been built being prepared in Wonderware Intouch Machine Edition (ITME) environment to configure and manage the local controllers of laboratory cabinets. As a showcase of the designed system, an example of residential autoconsumption has been emulated. Differences are pointed out with simulations by a comparative analysis between the showcase emulation and simulation in Matlab 2014a. Finally, a business idea is proposed with the active participance of the system and of the laboratory. The microgrid and the developed SCADA are based on the needs of forthcoming projects of the Institute.

Keywords: microgrid, SCADA, ITME, HMI, laboratory work, autoconsumption, emulation, simulation

Table of Content

ETSEIB

List of figures

List of tables

ETSEIB

Glossary

Abbreviation:	Explanation:
AC	Alternating Current
ANSI	American National Standards Institute
B2B	Back to Back or Business to Business
CAN	Controller Area Network
CERTS	Consortium for Electric Reliability Technology Solutions
CSC	Current Source Converter
CSI	Current Source Inverter
CW	Command Word
DC	Direct Current
DESI	Decentralized Energy Systems of India
DER	Distributed Energy Resources
DES	Distributed Energy Storage
DFIG	Doubly-Fed Electric Machine
DG	Distributed Generation
DS	Distributed Storage
DSO	Distribution System Operator
EMS	Energy Management System
EV	Electric Vehicle
HMI	Human-Machine Interface
I&C	Industrial & Commercial
IP	Internet Protocol
IPP	Independent Power Producers
IREC	Catalonian Institute for Energy Research
ISO	International Organization for Standardisation
KERI	Korean Energy Research Institute
NEDO	New Energy and Industrial Technology Development Organization
NTUA	National Technical University of Athens
OSI	Open Systems Interconnect
PLC	Programmable Logic Controller
PLL	Phase-locked loop
PMSG	Permanent Magnet Synchronous Generator
SCADA	Supervisory Control and Data Acquisition
SCIG	Squirrel Cage Induction Generator
SOC	State of Charge
SW	Status Word
TSO	Transmission System Operator

1) Introduction

“Think globally, act locally!”

/Patrick Geddes/

“The people who are crazy enough to think they can change the world are the ones who do.”

/Steve Jobs/

The electricity systems of the 20th century were mainly ruled by centralized, mostly fossil power plants. Nevertheless, the falling price of local renewable solutions (e.g. photovoltaic panels), and the rise of electric vehicles (EVs) make the grid more decentralized. [1] By smart metering, the Supervisory Control and Data Acquisition (SCADA) systems of decentralized consumption, production and storage can create well-functioning microgrids, systems, which may work independently from the central grid. [2]

“Think globally, act locally!” says the motto above, and microgrids truly empower local solutions in the energy sector. They are crucial for providing affordable, accessible and reliable electricity for all, fighting against energy poverty. SCADA systems are the basis of numerous, potentially disruptive business models in the energy sector. Amongst others, peer to peer energy trading, or optimized EV charging can create benefits for end-customers, as well as help energy utilities to stabilize the grid and to consume more sustainable energy. [3]

This master thesis primarily deals with the development of a microgrid SCADA system in laboratory conditions. The exact goals are listed in Chapter 1.1. The position has been posted on the CommUnity Platform by InnoEnergy in November 2017. After getting selected, the work has been conducted at the Catalonian Institute for Energy Research (IREC) from February to June 2018.

1.1. Aim and purpose, scope of the work

The general objective of the project is to create a SCADA system for the microgrid laboratory prepared for running emulations and to be further developed. The pre-defined SCADA system requirements received from IREC are exposed in Chapter 3.1.

More specifically, the following sub tasks have been tackled in this work:

- Describe the fundamentals and analyse the state of the art and market trends of microgrids. The main conclusions of this task are detailed in Chapter 1.2 and 1.3.
- Describe the electrical and communication schemes of the laboratory. This task is detailed primarily in Chapter 2.
- In Chapter 3 the core of the project, the SCADA system is presented. The SCADA is built for a laboratory microgrid according to the pre-defined functional requirements described in Chapter 3.1. All the details of the SCADA are linked as appendix to the corresponding section.
- Create a showcase demo of the developed SCADA, through which the system is tested, and the properties of an emulation are compared to computer simulation. This task is detailed in Chapter 4.
- Propose an innovative idea and develop a business model related to the showcase experiment and decentralized power systems. This task is detailed in Chapter 5.
- Finally, conclusions of the work are made in Chapter 6. These include the mapping of challenges and their solutions throughout the work, and directions for future development.

The main professional benefits for the master thesis student have been gaining experience and competence in sustainable energy solutions, SCADA programming, communication mechanisms, control techniques, business analysis and structured, collaborative work and documentation.

1.2. Fundamentals and state of the art of microgrids

A microgrid is defined as a local, potentially autonomous, low-voltage distribution network. As one of the most referred source of information in the topic, the U.S. Department of Energy describes, it is "a group of interconnected loads and distributed energy resources (DERs) with clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid and can connect and disconnect from the grid to enable it to operate in both grid-connected or islanded modes." [4]

The most important values of microgrids lie in the decarbonisation of the electrical network, and in the electrification of currently not-electrified areas. Many microgrids involve independent, local and sustainable energy resources, which are important segments of a universal, and sustainable electricity access. [3] The importance of microgrids has been growing in the past decades. This can be explained by two factors:

- Significant improvements in several Distributed Energy Resource (DER; e.g. photovoltaic panels) and Distributed Energy Storage (DES; e.g. lithium-ion batteries) technologies give business opportunities in the developed regions [5],

- The international fight for the electrification of rural areas in developing countries (e.g. sub-Saharan Africa, developing Asia). [6]

1.2.1. Value propositions of microgrids

In terms of grid-connections, there are two types of microgrids: grid-tied and isolated ones.

- Grid-tied microgrids, which may disconnect from the grid and operate in an islanded mode. The point where the microgrid is connected to the grid is called Point of Common Coupling (PCC). From the operator point of view this type of microgrid seems like a single controllable entity of the grid. [7]

- Isolated microgrids, which are used as the separated electricity networks where the connection to the grid is hardly achievable. It is expected that 50-60% of the additional capacities for a universal electricity access by 2030 would be supplied by these types of microgrids. [3]

The installation of grid-connected microgrids have several value propositions for the different stakeholders:

- For the end customers, they provide increasing reliability (backup in case of emergencies, and blackouts) by bringing close the generation to the consumption points. Moreover, they provide better power quality, higher energy independency, and they give opportunities for local energy community projects. Moreover, microgrids can decrease the cost of energy of end customers in long term. [7] [8]
- For grid operators (DSOs and TSOs) the main value proposition lies in the decreasing distribution and transmission losses and increasing energy efficiency. The installation of microgrids increases the penetration of renewables without the costs of redesigning the whole distribution network. Moreover, there are several examples of system flexibility improvements, i.e. better regulation of frequency and voltage, thanks to smaller, more easily operable grids. [5] By having various groups of local networks, operators may also use them as reserve and back-up generation for the network. [9]
- For governments microgrids increase national energy independency, as well as distributed energy generation increases the general protection of a grid against accidents and cyber-attacks. [7] Moreover, the changes caused in the power sector

enables new type of businesses, new type of jobs, which increases national economic growth, and ultimately makes a country more competitive. [7]

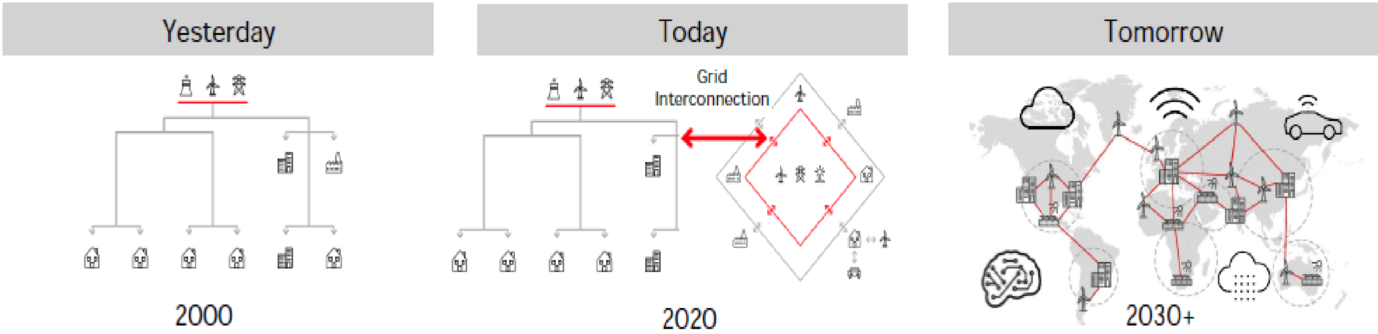


Figure 1: Electricity systems of yesterday, today and of tomorrow [26]

Microgrids fit in the trend of a radically changing energy economy and ownership structure. By the expansion of distributed energy systems, the power sector is getting closer to the customers. [3] Figure 1 shows this trend of the decentralization of power systems. This process incentivizes local communities taking over their energy management, saving money, as well as reducing their carbon footprints. [7] Grid-tied microgrids with DES may take advantage of electricity price variation and can decrease their operational costs by the smart utilization of renewable generation. There are numerous opportunities behind the idea of energy purchase of low-cost time frames, and of selling excess energy of a microgrid when prices are expensive, often in a peer-to-peer way. Peer-to-peer contracts further increase the upper-mentioned trend of operation without a central authority.

1.2.2. Challenges and state of the art research of microgrids

Besides the rapid growth and the upper-mentioned advantages of the technology, it faces several challenges to be solved [7] [8] [1]:

- Distributed Generation and a decentralized power system needs stabile, safe, and fast communication systems.
- Photovoltaics and Energy Storage Systems are based on inertia-less technologies. As traditionally the inherent stability of the public grid is based on rotating masses, the growth of microgrids gives stability issues to the grid.
- Numerous times energy policies do not keep the pace of technological development. There is still a lot of optimization work to be done in the field.
- Producing sufficient power quality and stability, especially during transition of the PCC gives control challenges.

Control agent approaches can be distinguished to two types: centralized configuration with a head agent, and multi-agent systems with several decision-makers, e.g. the SCADA and the EMS (Energy Management System). [10] In both cases microgrids are usually designed by considering the plug-and-play concept, meaning that the components can be changed to others without the need to re-engineer the whole system. [11] Two mentioned examples for energy managers are the SCADA and the EMS. Usually, the SCADA is responsible for communicating and supervising all devices, as well as directly controlling them and perhaps storing historical data in short term. [12] The EMS has a hierarchically higher role of control, taking into consideration long term tendencies, often weather forecasts and energy prices with energy analysis tools. [13]

These upper-mentioned challenges with various generation and load designs are studied internationally from laboratory to pilot-scale conditions. Project locations include the USA (CERTS - Consortium for Electric Reliability Technology Solutions), South Korea (KERI – Korean Energy Research Institute), Japan (NEDO – New Energy and Industrial Technology Development Organization), and several European countries (NTUA – National Technical University of Athens; IREC – Catalonian Institute for Energy Research). [7] The process of idea to full-scale testing can be observed in the American CERTS project, which mainly focused on generation-load control techniques. [11] The development of CERTS can be distinguished to three phases: simulation of the control idea in a computer; laboratory-scale test microgrid; and finally, the establishment of a pilot project and full-scale testing. Currently, research at these facilities focuses on the following topics [14] [15]:

1. The technological development of more efficient and affordable DER and DES. In these fields the competition is high. For example, cheap and flexible perovskite solar cells could revolutionize DERs, as well as NAS (sodium-sulphur) batteries would highly improve DESs. [16] [17]
2. Market participation and control strategies of microgrids, which are crucial for the wide acceptance of the technology. These control designs are usually made separately for grid-connected and islanded cases. Ultimately best control does not exist, there are different proposals for single agent, and for multi agent microgrid controls. [18] [19].
3. The detection of islanding phenomenon is crucial for keeping the stability and safety of both the public network and of the microgrid. Detection methods include local active (i.e. based on the response of the system for impulses), local passive (e.g. constant quality measurement of the power), hybrid (the combination of these two) and remote methods (e.g. power line carrier communication). [20]
4. With the introduction of microgrids, fault current treatment technologies of the power network would need redesign. The reason behind is that the assumptions of conventional treatments of fault currents and powers, i.e. fault currents decrease downstream, may not always comply because of the effect of the DERs and DESs. In this type of research, new types of relays are designed. Moreover, the utilization of

machine-learning algorithms is also explored for the changing conditions. [21] The economic and technical optimization for such solutions always need to be conducted after the analysis of local opportunities. [14]

5. Microgrid energy management, which can be central, decentralized, or distributed. In case of central control, one or more agents optimize the operation based on power demands, system stabilization, efficiency-maximization etc. Decentralized solutions offer higher flexibility for the local controllers which are responsible for most of the decisions. Nevertheless, the components of a distributed system are interconnected and communicate more with each other to achieve a pre-determined goal. As this optimization is many times rather complicated, several interesting proposals can be found in the literature, e.g. based on game theory. [22] Nevertheless, there are parallelly existing solutions currently on the market. [23]

Microgrids imply the increasing importance of communicational technologies in the energy sector, and the uprising trend of decentralization. For the better management, acceptance and transition to decentralized technologies, there would be a need of seamless standardized procedures, such as IEC 61850 and universalized components. This way, knowledge would be shared more efficiently and the efficiency of microgrid research would increase. [14]

1.3. Market drivers and trends of microgrids

1.3.1. Global trends of the microgrid market

The installed capacity of microgrids has been rapidly increasing in the past years up to over 2 GW. [24] The global microgrid market is currently estimated to be around US$ 12-17 billion and is expected to double by 2023. [24] [25] Based on these data, the market can be considered as an emerging market, which have not reached its maturity and full potential yet.

As it can be seen on Figure 2., the ruling regions of the technology are mainly North America and Pacific Asia. Amongst others, it is due to the higher density of islanded and not electrified areas. This supposition is also supported by the shares of the market segments seen on Figure 3. According to an independent research based on IEA, UNDP and UNIDO, to reach universal electricity access, in the emerging Pacific ASIA, Africa, and Latin America, over 35% of the electricity consumed would come from islanded microgrids by 2030. [6] This data indicates that the core for the growth of the microgrid market would come from these regions and market segments. In such islanded, and many times extreme (e.g. mines) conditions, the principal competitors technology of microgrids are diesel, and oil-based solutions. Hence, as many studies point out, the variation of oil price does affect the spread of micro networks, which ultimately makes predictions uncertain. [26]

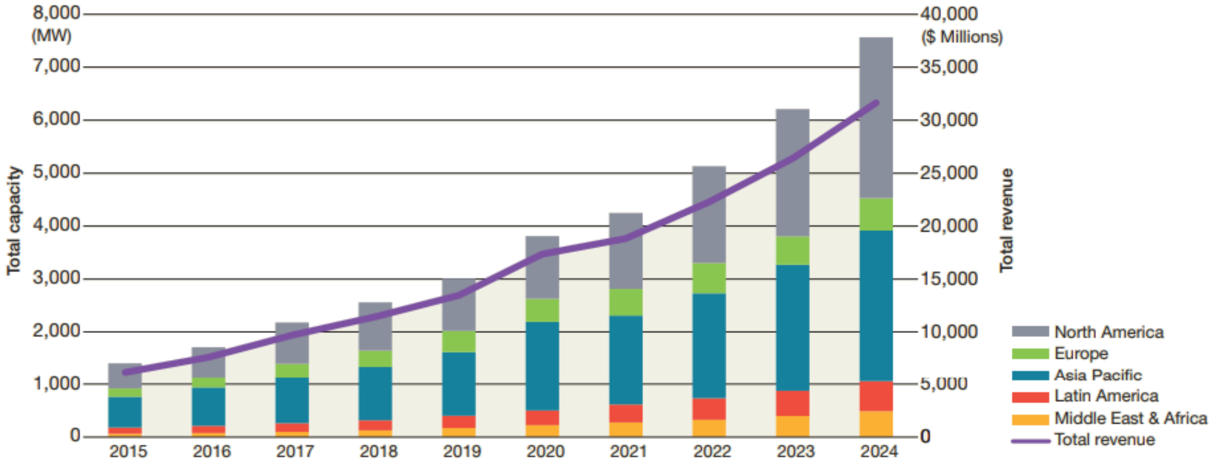


Figure 2: Total microgrid capacity and revenue by region, 2015-2024 [22]

1.3.2. Key drivers and market segmentation of the microgrid market

There is high research effort shown in the literature about the definition drivers and elements of the growth, as well as segmenting the customers of microgrids. According to the market analyser companies Visiongain Ltd. and IMARC Group, the driving forces include the growing adoption of renewable energy, mass electrification, incentives against climate change and the increasing demand for more efficient power systems as opposed to the aging infrastructure. On the other hand, the high cost of installation and the general lack of technical expertise result to be challenging factors. [27] [25]

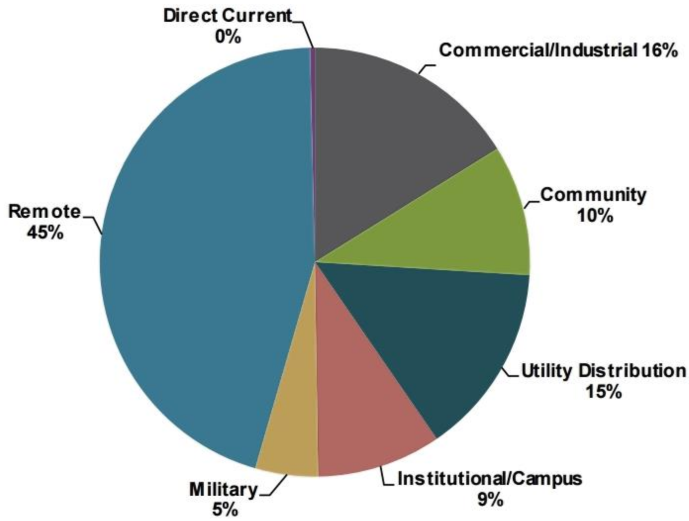


Figure 3: Shares of the market segments [27]

Table 1 shows the market segmentation and the proposed values for each group. [28] In the matrix, green boxes mean main drivers, and yellow ones stand for secondary drivers. The values mentioned in Chapter 1.2.1 are grouped as follows:

- social, i.e. the electrification of not electrified areas,
- economic, i.e. cost saving,
- environmental, i.e. the reduction of the ecological footprint,
- operational, i.e. higher independence and power security.

The customer market can be segmented to the following groups: utility distribution of microgrid, remote communities, industrial & commercial sector (I&C), military/defence, energy communities and institutions/campuses. Institutional projects have been the main initial pioneers until recently. This has been changing for the remote community and for the I&C segment, giving currently more than half of the microgrid-related projects. [29] For the review of the various value propositions, two example project cases are described below: one for the I&C segment, and another one for the remote community segment.

Table 1: Market segment and driver matrix of microgrids [28]

		Social	Economic	Environmental	Operational	
Segments	Typical customers	Access to electricity	Cost savings	Reducing ecological footprint	Fuel independence	Uninterrupted supply
Utility distributions	Utility, IPP		X	X	X	(X)
Remote communities	Utility, IPP, Governmental development institution, development bank	X	X		X	
Industrial and commercial	Mining Company, IPP, Oil & Gas company, Data centre, Hotels & resorts, food & beverage	(X)	X	(X)	X	X
Defence	Governmental defence		(X)	(X)	X	X
Communities	Utility, IPP			(X)		X
Institutions and campuses	Private education institution, IPP, Government education institution		(X)	X		(X)

As an example project of the I&C segment, a project in Burkina Faso is under closer review. [30] The IAMGOLD Essakane gold mine microgrid project has been commissioned at the end of 2017. The main value proposition of this microgrid was a completely autonomous, local energy system for a mine located far away from the public grid. The supply is based on 55 MW oil power plant supplemented by 15 MW PV. The new PV installations decrease the fuel consumption and increase the fuel-independence of the Company. Moreover, this hybrid solution also decreases CO2 footprint of the mine by 18 500 tons per year, giving a green perception from society to the Company. [30]

For remote communities, where microgrids are the sole solution for electricity access, typical example projects can be found in Asian countries. Such countries usually put high policy and incentive effort, backed by international directives and grants, to achieve mass electrification. [6] In India, the non-profit organization Decentralized Energy Systems of India (DESI Power) is a main designer and installer of usually biomass-based microgrids. [31] The mission of the organization is to decrease the “gap between haves and the have-nots”, including access to electricity. The vision of DESI is achieving it by decentralized, environmentally clean, and affordable energy services. The financing of DESI Power is via international grants and individual funds. Nevertheless, DESI projects often merge commercial and residential customers. This way the Company can stabilize itself financially, and still move towards its mission. [32] Solutions provided by DESI power are often complex and the needs, possibilities of the particular customers are reflected on the particular product. Such projects are often managed on low budget and are good examples for microgrids without the latest wireless control and metering systems.

1.3.3. Regulatory aspects of grid-connected microgrids

Several research challenges described in Chapter 1.2.2 also imply actions from regulators. The laws and directions are normally made for centralized power utilities, and as such, do not consider local projects. For example, the exact conditions, procedures of the switch between islanded and grid-connected mode are challenges political decision-makers currently face with. Moreover, franchise rights of utilities, obligations of the stakeholders of a project and of the grid, taxation, and incentives for national and international goals are examples what regulators should address for the safe and stable integration of microgrids to the public grid. [33]

In case of urban networks, e.g. institution and community-based projects, the regulatory impact is especially important, since these systems are mostly grid-connected and have high dependence on the rights and obligations with the local DSO. In the occidental world, energy markets are usually unbundled, meaning that the four energy market activities, i.e. generation, transmission, distribution and retail, must be operated by a separate actor. The original goal of the unbundling concept is to increase competition and decrease costs for final customers. Nevertheless, according to numerous research, it has a counter effect in case of microgrids. [34] Hence, these mini grids of the public network numerous times can be exempt from unbundling policies. [30] According to the third Energy Package of the European Union member states can decide about the unbundling conditions for “integrated electricity undertakings serving less than 100,000 connected customers or serving small isolated systems”. [35] This directive is a typical example of how the transition towards microgrids can be supported in a liberal energy market.

2) Laboratory environment in IREC

2.1. General introduction to the laboratory

The Energy SmartLab Laboratory of IREC, Barcelona has a history of constant development in the past years. The low voltage laboratory of 200 kVA aims to develop cutting-edge smart energy solutions including DERs. Amongst others, the laboratory is used to investigate challenges in the distribution networks, management and control of micro networks and to foster the integration of renewables and of electric vehicles. It has a unique infrastructure with real and emulated equipment, offering a great flexibility, modularity and configurability for numerous types of projects. [9] [36]

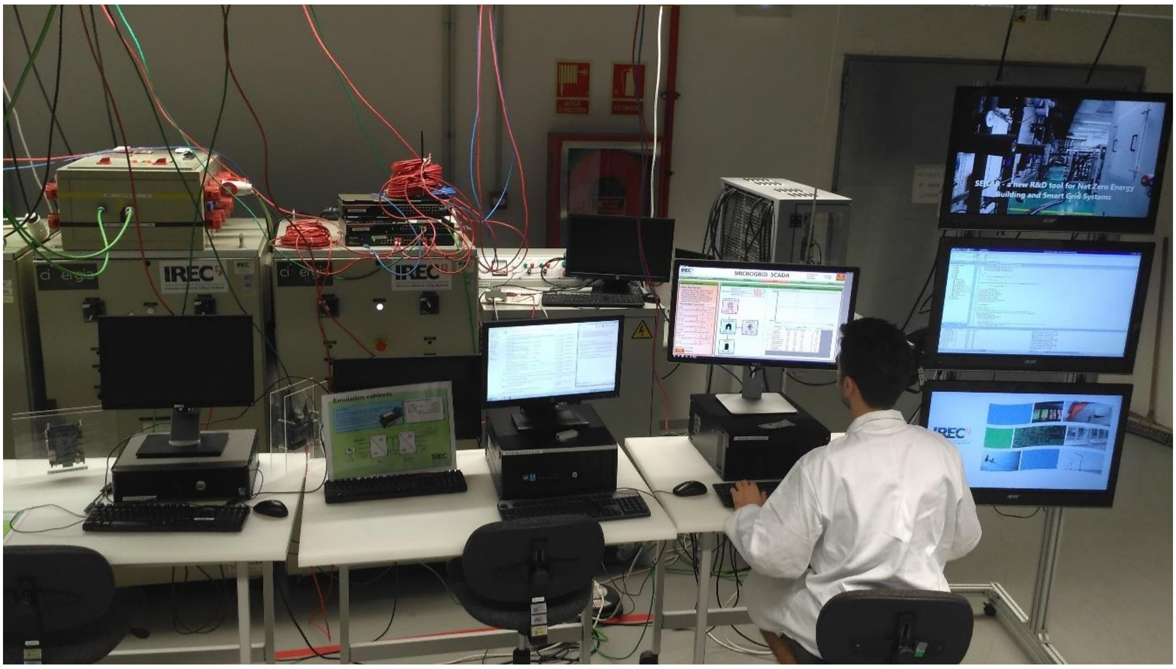


Figure 4: IREC's SmartLab with the emulation cabinets, the EMS and the SCADA

By using both real and emulated cabinets, SmartLab consists of several types of configurable units of distributed generation (DG), distributed storage (DS) and of consumption. All elements are modular and are provided with a Local Controller (LC). These local controllers are communication boards to each cabinet of the laboratory. They are connected to manager computers, such as to the EMS or to the SCADA system under development. The development of this latter system is the core goal of this project. Further Chapters provide more details about all elements, including their electrical, and communicational architectures.

2.2. Electrical architecture: real and emulated technologies

The laboratory microgrid at IREC currently operates with 5 emulation cabinets, 4 semi-emulated wind turbines and 3 real elements.

- Real elements are currently batteries and a supercapacitor in the laboratory. Chapter 2.2.1. gives further specification of these devices.
- Emulation cabinets are devices which can be configured to generate or to consume real power in the microgrid laboratory under configured conditions. [9] Chapter 2.2.2. elaborates more on the functioning of these cabinets.
- Semi-emulated wind turbines have real motor generators of those which can be found in windmills. They have a rotating mass, and hence, can be used for output fluctuations of wind generators. [9] Nevertheless, as currently they are not fully integrated yet, they are not further discussed.

Figure 5 shows the electrical scheme of the components, connected in a parallel way. The main advantage of having a potentially closed system is that public grid is only needed for balancing the losses. The exact list of all elements can be seen in Appendix B, and the ones being important from the current SCADA development point of view are listed in further chapters.

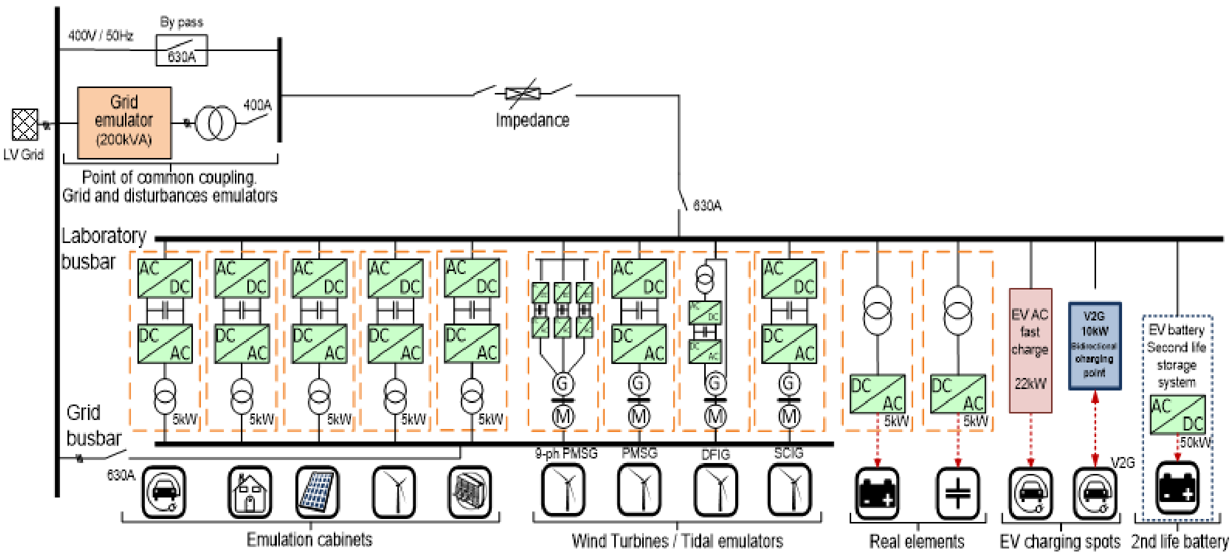


Figure 5: Electrical architecture of the microgrid

2.2.1. Real systems

There are overall 3 real elements currently in SmartLab: an ultracapacitor, a SAFT lithium-ion battery and a second life lithium-ion battery. Further elements, such as a flywheel are also available, however, are currently not in use. All presently available real elements are based on electrochemical storage technologies, i.e. electrical energy transforming to chemical energy or chemical energy to electrical. All these devices are equipped with power electronics for their control, facilitating their connection to the microgrid. Each converter is connected to its corresponding LC which gives and receives information via CAN from the power converters. Table 2 shows the technical specifications of the Saft battery and of the ultracapacitor.

The connection of the Saft battery to the microgrid is solved by an AC/DC and a DC/DC converter, seen in Figure 6. Firstly, the DC/DC converter is used as a step-up converter up to DC 700 V. This DC voltage is then converted to the operating AC 400 V 3-phase line voltage of the microgrid. The DC bus between the two converters is modelled as a capacitor.

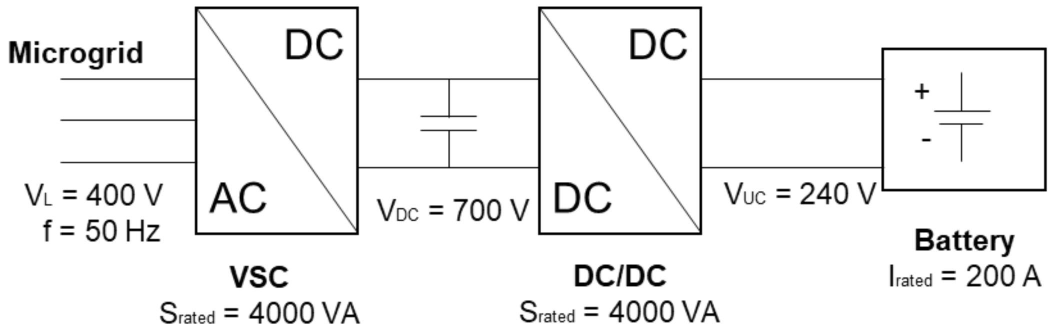


Figure 6: Saft Battery connection to the microgrid

Ultracapacitors are based on the surface reactions of two electrodes, having exceptionally high efficiency due to high surface of the used activated carbon. [37] The 57 Wh, 10 kW ultracapacitor cabinet has a response time of less than 1 second and a high-power output. Moreover, ultracapacitors may have thousands of charging-cycles and high lifetime. [38] Based on these advantages, ultracapacitors are usually used for emulations with the goal of grid regulation. [37]

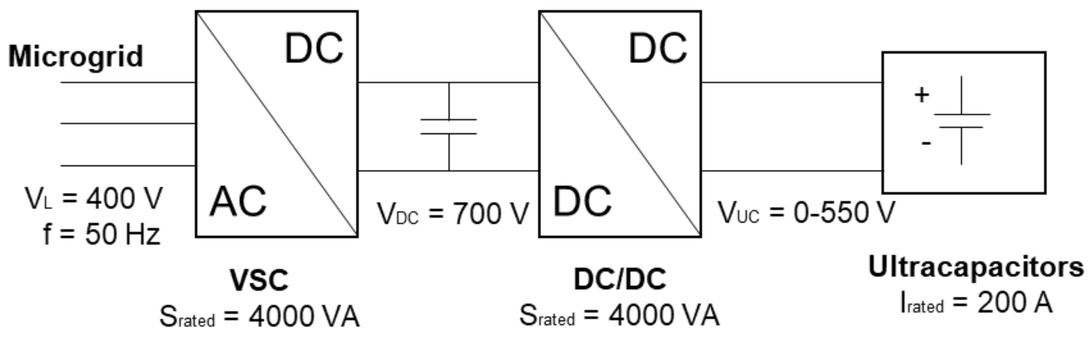


Figure 7: Ultracapacitor connection to the microgrid

Figure 7 shows the ultracapacitor connection to the microgrid. The control and connection technology are similar to the battery one, having an AC/DC and a DC/DC convertor in series with the capacitors.

Table 2: Technical specifications of the considered real elements

Parameter:	**Lithium-ion battery:**	**Ultracapacitor cabinet**
Stored energy:	20.000 Wh	57 Wh
Operating voltage:	189-227-254 V	250 V – 550 V
Rated discharge current:	200 A	20 A (peak: 200 Apk (<1s)
Rated charge current:	34 A	n.d.
Rated power:	150.000 W	10.000 W
Series/Parallel combination:	10*(7/2)	35/1
Capacity:	82 Ah	1.65 F

2.2.2. Emulated systems

Emulated cabinets are hardware emulated equipment, which have two bidirectional AC/DC converters in B2B (back to back) connection. This way the elements can inject or consume power, depending on the set configuration. [39] These cabinets can reproduce the electrical behaviour of different elements, such as of renewable generation, of residential consumption, or of energy storage systems. An emulation is differentiated from simulations, as well as from real elements by the following advantages:

- Comparing to sole simulations, it has the advantage of real power flow during an experiment, giving opportunities for more realistic testing and management algorithms.

- As opposed to a real element, an emulation, e.g. photovoltaic emulation, gives the benefit of higher flexibility and weather independence. With emulation cabinets, several, normally uncontrollable phenomena, i.e. weather fluctuations, can be studied under controlled conditions. [40]

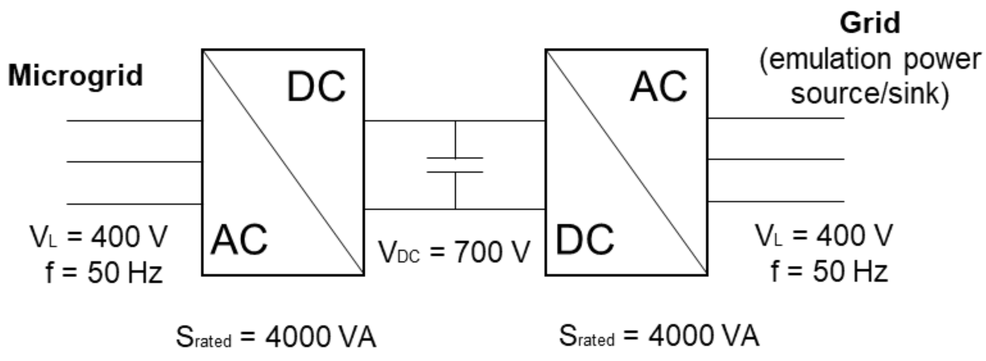


Figure 8: Emulation cabinets to the microgrid- power converters in B2B connection

Figure 8. shows the back to back connection of the two AC/DC converter. The power converter on the microgrid side, i.e. the emulator, is programmed for setting a maximum power, which can flow through the DC bus between the two converters. The other, grid-side converter is the active front end (AFE), which is responsible for power injection towards the DC bus or to the public grid, as well as conducting constant measurement of the equipment. [41] In case of power injection to the microgrid, the emulator enables the power flow from the grid towards the DC bus. In consuming mode, the node dissipates power from the DC bus which flows towards the grid. This kind of technology makes it possible to emulate any type of power source or consumption.

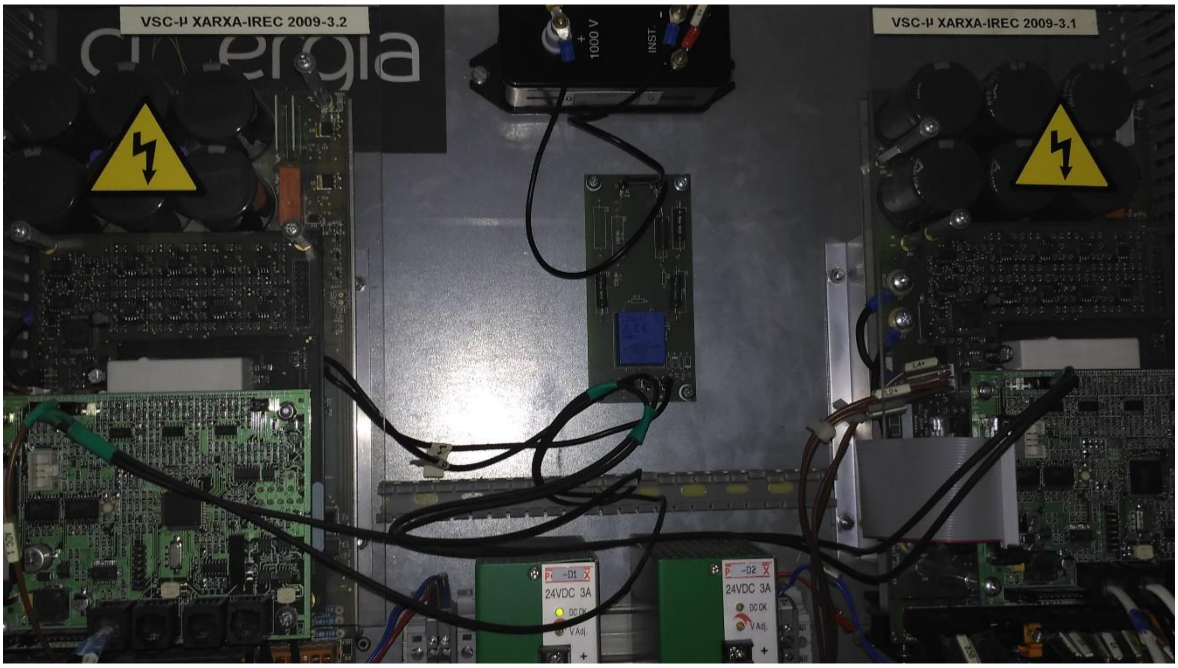


Figure 9: Picture of the two power converters in B2B connection

Table 3: Technical specifications of emulation cabinets

Parameter:	Value:
Rated power:	4.000 VA
Operating voltage:	3 x 400 V
Availability:	5 cabinets
Operation modes:	Stored curve, storage and set point following

In IREC's SmartLab there can be 5 B2B-uX-4kW emulation cabinet found. The two converters of each configuration are connected by means of a DC bus of 700 V. The exact parameters of the converters can be found in Table 3.

As the control of the converters is important to be fast and reliable, each of them has a corresponding LC, which are responsible for all the communications with the them via CAN protocol. In the LCs several emulation types are previously configured. Although this list is expected to further expand, currently the following consumption or generation of DERs are prepared:

- controllable battery,
- controllable wind turbine,
- non-controllable wind turbine,
- controllable solar generator,
- non-controllable solar generator,
- controllable consumption,
- non-controllable consumption,
- and tracking tester.

In this matter, controllable or not controllable makes the difference between those which may follow external control from Modbus and those which solely follow the loaded profile in the LC. This functioning is further explained in Chapter 2.3.2.4.

2.3. Communicational and management architectures

The control and management system of the laboratory is hierarchical, running on three levels and with two types of protocols. These two protocols are Modbus TCP/IP and CAN (Controller Area Network).

Modbus TCP/IP is a free communication protocol at the application level of the ISO/OSI model (ISO/IEC 7498-1). [42] It is typically used for server and client types of communications, such as in the field of control devices and data gathering systems. [42] Speeds of Modbus TCP/IP type of communication typically range up to 1.25 Mbit/s. [42]

CAN refers to a protocol being present in both data (ISO 11898-1) and in the physical layer (ISO 11898-2) of the ISO/OSI model. [43] It is cheap, but durable, in which electronic control units may have one single interface, making the messages more transparent, and less wired. [43] This protocol is used widely in industries with sensors and constant communication of measurements. CAN speeds range up to 1 Mbit/s. [43]

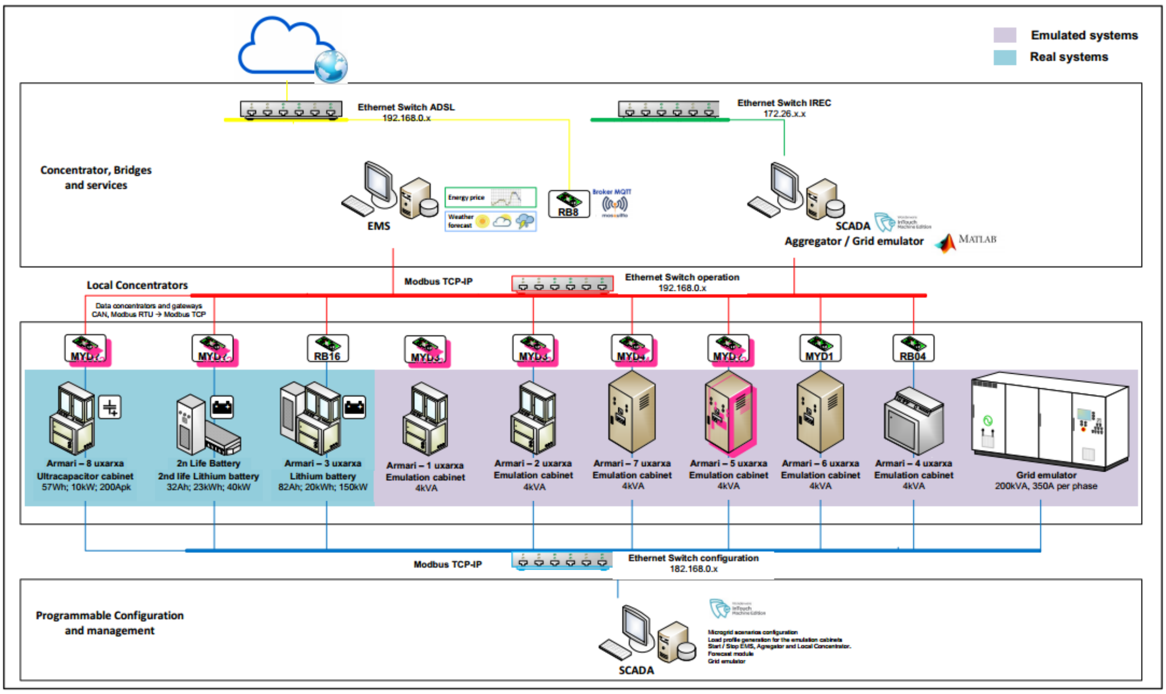


Figure 10: General communication architecture of the laboratory

No cabinets are allowed to be connected to any networks leaving the laboratory. The only computer in the laboratory which is connected to IREC's network and to the internet is the SCADA. The explanation behind is to facilitate remote access of the SCADA, and hence, being able to supervise the laboratory physically not being present. For the sake of getting real-time information, the EMS is also expected to connect to the internet, nevertheless, would run on a separate line.

As upper-mentioned, the communication and management system of the laboratory is hierarchical, and can be distinguished to three parts [44]:

1. The upper layer: the supervisory and control computers (e.g. EMS or SCADA);
2. The middle layer: the LCs;
3. The bottom layer: the power converters.

Communications between the top and the middle layer are executed via Modbus TCP/IP over Ethernet cables. Figure 10 shows the basic communication schemes of the laboratory. Those highlighted in red are currently in repair, modification or update. Originally, the operation and the configuration modules (see Chapter 2.3.3) would run on different local lines. Using two different lines for the communications has the benefit of avoiding unrealistic communications, as well as keeping the network safer. On the operational network (192.168.0.x) the grid emulator is not connected, however, on the configuration line (182.168.0.x), it is. Nevertheless, currently the operation line is the sole local network used in the facility.

Between the middle and bottom layer CAN is used. The LCs gain information from the cabinets via CAN, send commands, and power setpoints to be reached. During operation, they have two types of mode: manual and auto. If the LC is in manual mode, then it only switches power setpoint upon a specific command. If it is in auto mode, then setpoints are sent in each second.

In the following chapters the functioning of each communicational layer is described in a bottom up approach.

2.3.1. Bottom layer: power cabinet controls

Apart from the general measurement data, the state of the converter is the most important information a cabinet produces. The LCs receive information about the current state of the converters in the Status Word (SW) messages and they can change their state by sending Command Word (CW) via CAN. The following points describe the physical meaning of each SW:

- **SW: "Standby state" (1)**: a cabinet in standby state if it is powered and ready to precharge;
- **SW: "Transition towards precharging state" (2):** a cabinet is in transition state between receiving the command for precharge and starting the process;
- **SW: "Precharging state" (3):** when a cabinet is in precharging state, its DC voltage slowly increase until 90-95% of the operating voltage. During precharge the current flow is limited in order to avoid overcurrents;
- **SW: "PLLing state" (4)**: when a cabinet is PLLing (Phased-Locked Loop) state, its internal control system is getting tuned for the start of the emulation;
- **SW: "Operating state" (5):** in operating state the cabinet is ready to receive power setpoints. It may be switching, or may not, depending on the mode and on the commands received;
- **SW: "Stopped state" (6):** the cabinet is in stopped state if the internal algorithms puts in this state or if the LC sends the command for it;
- **SW: "Emergency stop state" (7):** the cabinet gets in emergency stop state if any internal algorithm, or CAN command says it so. Moreover, emergency stop can get executed manually on each cabinet.

As regards to CWs, the cabinets accept the following commands via CAN:

- **CW: “Emergency stop” (1):** sends emergency stop command. The emergency mode shall be followed by the reboot of the cabinet;
- C**W: “Precharge” (3):** sends the command to the converters to start their precharging sequence;
- **CW: “Start switching” (4):** sends the initial and the upcoming power setpoints of P and of Q;
- **CW: “Stop switching” (5):** sends stop command;
- **CW: “Ask for an update” (F):** sends a request for data update.

2.3.2. Middle layer: Local Controllers

2.3.2.1. General description

The LCs can read and write the necessary data directly on the power controllers via CAN, providing a common communication interface to the cabinets. Normally, LCs are placed on the top of the cabinet they are connected to. These devices need to cover all the needs, and at the same time, need to be cheap enough to be purchased for each equipment. Currently, there are two types of LCs in use, shown in Figure 11:

- the original ones are MYD-AM335X, which are currently being changed to a new type;
- the new ones are Raspberry Pi 3.

This change does not only imply physical change, but a new configuration program is being implemented as well. The comparison of the current LCs and configurations with the new LCs and configurations are summarized in Chapter 2.3.2.2.
The LCs are equipped with GNU/Linux and they are programmed in C++ to produce easy configuration and monitoring services for the cabinets. The power profiles are expected to be

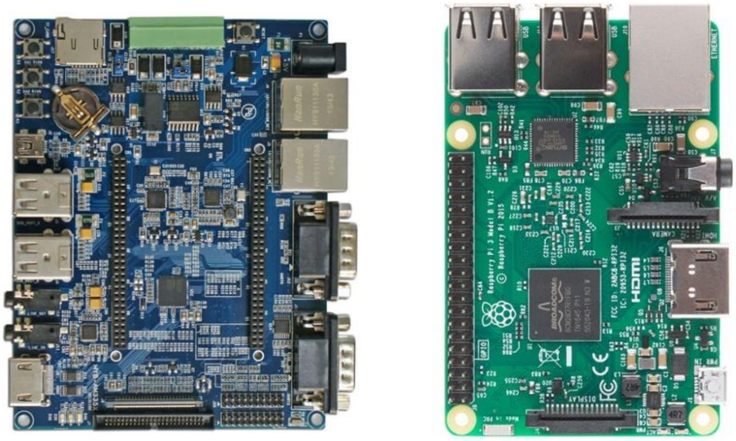


Figure 11: Photo of the old (left) and the new (right) LCs (MYD-AM335X and Raspberry 3 pi)

put in a specific folder, with a name *Profile_10_02.csv*, where the first two integers stand for the type code and the second two are for the power profile number. These profiles have up to 604800 lines, where each line stands for a second, meaning that the power profiles can contain the profile of a cabinet for a week. This solution facilitates easy power curve change for any emulation.

Figure 12 shows a screenshot of the user interface of the old LCs. The screen can be distinguished into three zones: the input zone on the upper-left, the state zone on the upper-right, and the event zone on the bottom part. The LC can be controlled by keyboard commands, each corresponding to a CW.

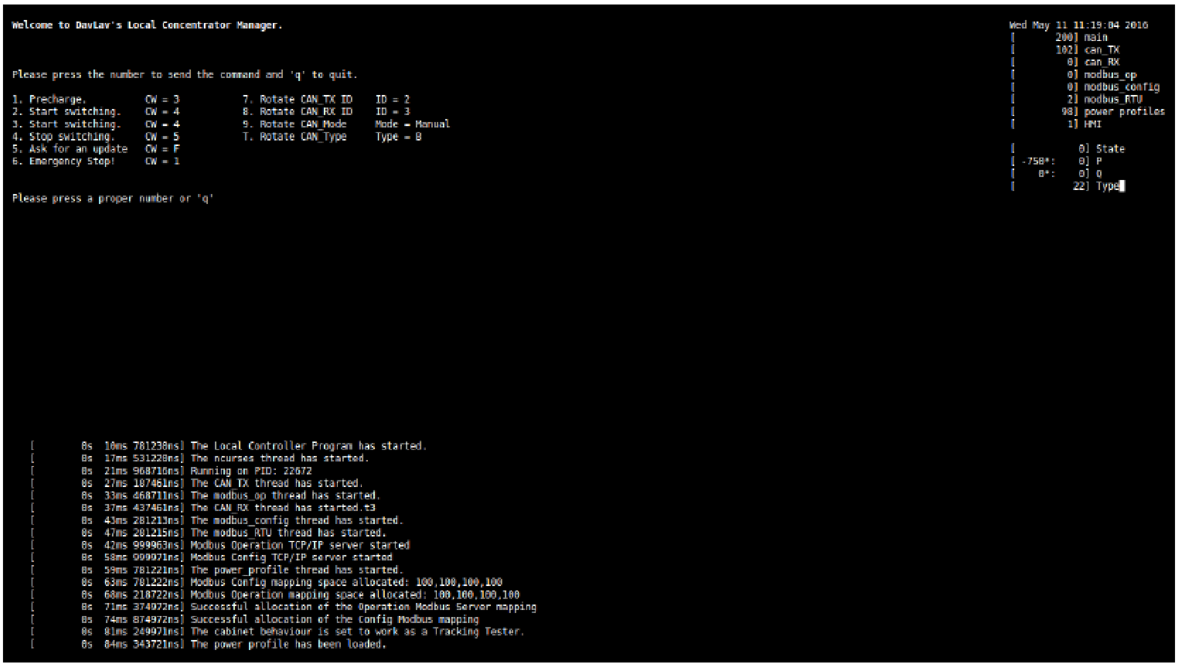


Figure 12: User interface of the old LCs

2.3.2.2. Comparison of old and of new configurations

Along with the replacement of MYD-AM335X development board, all LCs are currently being reprogrammed to cover all the upcoming new needs. The differences between the old and the newly developed configurations can be studied in Table 4.

Table 4: Summarizing table of differences between old and new LC configurations

	Old configuration	**New configuration**	
Development Board	MYD-AM335X and Raspberry pi 3	Raspberry Pi 3	
Configuration and monitoring module	Distinction by port 1500 and 1501	No distinction between the modules	
State Machine Diagram	Not known	Known, see Appendix A.	
Starting method	Particular method is needed, see below.	Profile is started by CW="start switching" (4) or by MB start==1.	
Modbus TCP/IP mapping	41 elements, see Chapter 2.3.2.3.	Old elements are complemented by 11 more, see Chapter 2.3.2.2.	
Type codes:	**Type code**	**New Local Type code**	**Controllable**
Battery (real)	2	1	Yes (always)
2nd life Battery	2	2	Yes (always)
Ultracapacitor	-	3	Yes (always)
Controllable Battery (emulated)	1	23	Yes
Controllable Wind Turbine	3	22	Yes
Non-Controllable Wind Turbine	4	22	No
Controllable Solar Generator	5	21	Yes
Non-controllable Solar Generator	6	21	No
Controllable Residence	9	20	Yes
Non-controllable residence	10	20	No
Tracking Tester (not implemented currently)	22	30	Yes

Apart from the CAN ports, the LCs are equipped with two Modbus TCP/IP servers: one for the operation (port 1500) and one for the configuration (port 1501) module (on networks 192.168.0.x, and 182.168.0.x). The differences between these two modules are described in Chapter 2.3.3. Although these two channels were originally thought for safety reasons, currently solely the switch 192.168.0.x is under operation. Nevertheless, the two ports are still applicable in case of the old configurations. For the new configuration, solely port 1500 is planned to be used.

Although the state machine diagram of the old LC is not known, the new configuration algorithms are planned with a diagram in mind. This is further elaborated in Chapter 2.3.2.4.

Although the new configuration will allow an easy method for start and management, the old configuration has several restrictions, which force current users to follow a strict method for the start of an emulation:

1. Powering up the cabinet.
2. Precharge management from the LC. Prepare the cabinet to reach state="Operating" (5). Make sure, that the mode of the cabinet is in manual.
3. From the SCADA change the type and the profile number code to 0.
4. In order, allow the SCADA changing the profile start minute, the profile number and the type of the emulation.
5. By switching the LC mode from manual to auto, the cabinet starts switching.

After the emulation is done, follow this sequence to facilitate another one:

1. Stop switching by changing the LC from auto to manual mode.
2. Quitting the program with q, and ctrl+c. The controller gets disconnected from the Modbus TCP/IP network.
3. Restarting the program "thread15" on the LC.

Both LC configurations allow numerous types of emulations. The two configurations follow different types of convention. As it can be seen in Table 4., new configurations separate the controllable property from the type itself. This way the number of type codes is reduced. Moreover, in the new configurations, integers below 10 indicate real elements, integers over 20 indicate emulated element types. By this convention the user would obtain a better intuition when seeing a type code.

2.3.2.3. Modbus TCP/IP mapping

The data transmission between the LCs and the SCADA is executed via Modbus TCP/IP protocol. These are all the details on Modbus TCP/IP which may be presented in the SCADA. Their full appearance is described in Chapter 3.4.4.2. Table 5 contains the basic available information with its Modbus mapping of the cabinets. The exact specifications of all these mappings can be seen in Appendix C.

Table 5: Table of Modbus addresses, field names and explanations, old configuration

Address:	Field:	Comment:
0	Type of cabinet	The type of the cabinet; see Table 4 for more information.
1	Power Switch	1 if the cabinet is switching, 0 if not switching.
2	Maximum active power	The maximum active power in W the cabinet allows flowing through its DC bus.
3	Minimum active power	The minimum active power in W the cabinet allows flowing through its DC bus.
4	Maximum reactive power	The maximum reactive power in var the cabinet allows flowing through.
5	Minimum reactive power	The minimum reactive power in var the cabinet allows flowing through.
6	Active power setpoint	In case of controllable mode, this variable might overwrite the loaded profile. For more information, please refer to Chapter 2.3.2.4.
7	Reactive power setpoint	In case of controllable mode, this variable might overwrite the loaded profile. For more information, please refer to Chapter 2.3.2.4.
10	State	It shows the state of the converter before an error happens. Otherwise, CAN state = State....
11	CAN State	The SW received from the converter (refer to Chapter 2.3.1.)
12	CAN Error	It shows the error code when the converter fails. Otherwise, its value is 0. Please refer for error code meanings to Appendix E.
13	CAN read ID	CAN identificatory for reading messages. Its significance is out of scope.
14	CAN transmit ID	CAN identificatory for sending messages. Its significance is out of scope.
15	CAN mode	Manual (20) or automatic (10).
16	CAN command	The CW being sent. Please refer to Chapter 2.3.1.
17	CAN P	Active power injected in W.
18	CAN Q	Reactive power injected in var.
19	CAN read 11 iterations	Number of CAN messages received. Its exact significance is out of scope.
20	CAN Iac	AC current in mA.
21	CAN Vac	AC voltage in V.
22	CAN Frequency	Frequency of the grid in mHz.
23	CAN read 12 iterations	Number of CAN messages received by package 12. Its exact significance is out of scope.
24	CAN Idc	DC current of the cabinet in mA.
25	CAN Vdc	CAN Vdc DC voltage of the cabinet in V.
26	CAN Temperature	Temperature of the cabinet in ºC.

27	CAN read 13 iterations	Number of CAN messages received by package 13. Its exact significance is out of scope.
28	CAN Type	Its significance is out of scope.
29	CAN P setpoint	Active power setpoint sent to the emulator in W.
30	CAN Q setpoint	Reactive power setpoint sent to the emulator in var.
31 and 32	Modbus operation read iterations	Number of iterations of the main program every time a Modbus TCP/IP message is read. Its exact significance is out of scope.
33 and 34	Modbus operation transmit iterations	Number of iterations of the main program every time a Modbus message is written. Its exact significance is out of scope.
35 and 36	Modbus operation read write iterations	Number of iterations of the main program every time in case of read-write of Modbus. Its exact significance is out of scope.
40	File number power profile	The file number of the configured profile under the specified folder.
41	Initial minute in power profile	The initial minute of the profile from where the switching shall start.
42	Current minute in power profile	In case of a running emulation it shows the current minute of the profile.
43 and 44	Current iteration in power profile	Current iteration in power profile. Its significance is out of scope.
46	Samples per second	Samples per second in the power profile.
51	SoC	State of Charge in % in case of a battery or of a battery emulation.

Additionally to the upper-listed addresses, Table 6 shows the data expected to be transmitted when the new LC configurations are implemented. Their exact significances are generally out of scope, except for those which are used in the new state machine algorithm. The new state machine algorithm is exposed in Chapter 2.3.2.4.

Table 6: Table of additional Modbus mapping of the new configuration

Address:	**Field:**	**Comment:**
51	CAN State 2	State of the converter back.
52	CAN RX 2	CAN identificatory for reading messages in the second converter.
53	Initial iteration	Initial iteration of the power profile.
54	Profile Config	Variable to set new profile.
55	Controllability	Boolean indicating the device’s controllability.
56	Start switch	Variable to start switching.

57	Start profile	Variable to start profile.
58	Experiment reset	Variable to restart profile.
59	Stop profile	Variable to stop profile.

2.3.2.4. State machine of the new Local Controllers

The state machine algorithm is the core of the configuration of an LC. The algorithm largely builds on the cabinets' status and command word (SW and CW). As the LC receives the SW of the cabinet, it responses with a pre-programmed CW and perhaps other internal functions are run. The flowchart of the algorithm can be seen in Appendix A. Based on this state machine of the LC there are several actions planned in the SCADA. The plan for these actions is further described in Chapter 3.5.1. The following paragraphs describe the exact steps of the algorithm.

At the beginning of the sequence, in step one the algorithm checks if it has received an emergency stop command. If not, then goes to the next step.

In step two the LC algorithm checks if the cabinet is powered, but not precharged yet. (SW<"precharge" (3)) If not, then the algorithm goes to step three. If yes, then the algorithm checks the mode (auto or manual) of the LC. If it is in manual mode, then waits for further commands. If it is in auto mode, then the LC automatically sends the command for precharge.

In step three the LC algorithm checks if the cabinet is in PLLing state (SW="PLLing" (4)). If not, then the algorithm goes to the step four. If yes, then it checks if the MB variable *MB Profile_config* (normally controlled from the SCADA) gives the order to download the current profile type from the SCADA. If yes, then it does it so. If not, then it checks if the MB variable *MB start switch* (normally controlled from the SCADA) gives the order to start switching. If yes, then it sends the command to the cabinet to start switching.

In step four the LC algorithm checks if the cabinet is in Operating state (SW="Operating" (5). If not, then the program goes to step five. If yes, then it checks if the MB variable *MB exp reset* (normally controlled from the SCADA) gives the order to reset the profile. If this command is given, then it stops, gets reinitialized, and toggles *MB exp reset.* Afterwards, or if the resetting command is not given, the algorithm checks if the MB variable *MB start profile* (normally controlled from the SCADA) gives the order to start the profile. If yes, then the sequence is stopped CW="start switching" (4) is sent with active and reactive power setpoints of 0. If the command for *MB start profile* is not given, then the algorithm reads the next line of the profile. If the LC is not in controllable mode, it sends the next power setpoint as the minimum of the upcoming profile value and the maximum power allowed on the DC bus. If the LC is in controllable mode, it sends the minimum of the upcoming profile curve value, the maximum power allowed, and of the control setpoint received on MB. If the profile has ended, then *MB stop* is sent.

The last two steps, step 5 and 6 check if SW="Stopped" (6) or SW="Emergency Stop" (7), respectively.

2.3.3. Upper layer: central management and the role of the SCADA

According to the laboratory roadmaps, there are three main data managers and control devices planned in the laboratory: the aggregator, the EMS, and the SCADA. The initial development of this SCADA is the core of the project.

The orders to the LCs are given from the SCADA, as well as it monitors and gathers all the necessary information from the local elements. It may also act as a communication gateway between the EMS and the LCs. The EMS provides smart energy management based on real-time optimization of numerous data, i.e. history, prices, forecasts. The aggregator, as the hierarchically top manager may give restrictions, and define the flexibility which influence the behaviour of the network.

The microgrid is prepared to run several types of projects and scenarios. The different projects might need completely different setups, which in general are called demos. These demos are usually defined and parametrized from the SCADA. As it has already been mentioned before, and can be seen on Figure 10, there are two modules of the microgrid elements which need distinction: the configuration and the monitoring module. In configuration module all the important planning and arrangement of each element shall happen. In the monitoring module the LCs control the cabinets on power, and the LCs are supervised by the SCADA, which then communicates with other agents and may act as a manager. The following Chapters introduce what exactly configuration and monitoring means from each participant's point of view, when running an emulation.

2.3.3.1. Configuration module

The configuration module is mainly used during the establishment and arrangement of a scenario. Figure 13 shows the communication exchanges between the main central bodies under this module. In the configuration module the SCADA would define the mode of the emulation: manual or auto. This distinction gives different roles to the different elements of the network.

- Manual mode means that the SCADA and the whole microgrid work more independently from the EMS. The SCADA controls and monitors the microgrid, and it ignores any messages coming from the EMS. This mode gives higher flexibility for the operation; however, it is less suitable for automatic control algorithm based on energy price and forecast.
- Auto mode of the network changes the roles, and most of the control information derives from the EMS. It optimises and give commands to the controllable elements of

the system depending on the forecasts of price, demand and generation. In automatic mode the SCADA system would only behave as a gateway, communicating the EMS orders towards the LCs.

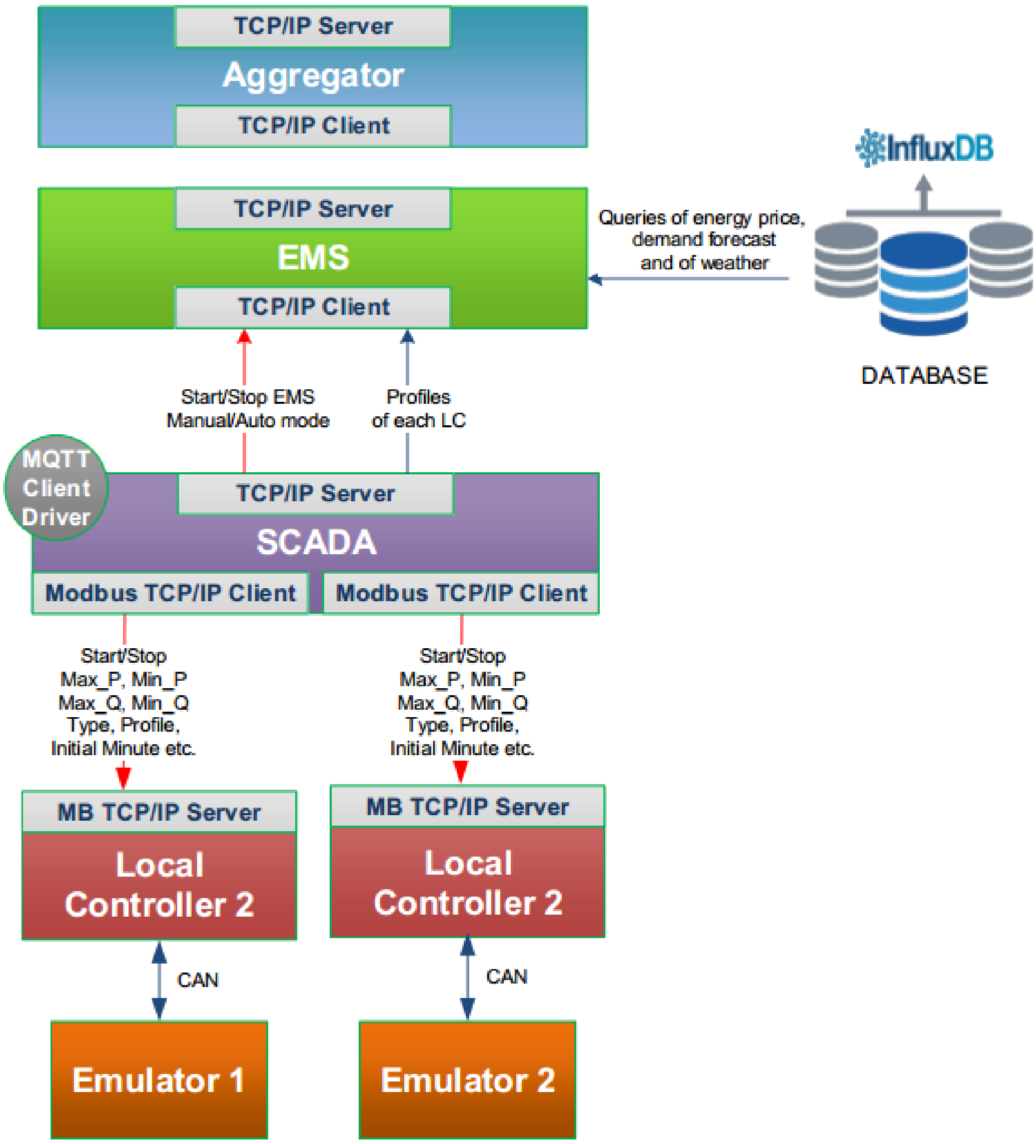


Figure 13: Information exchange between the participants in the configuration module

Apart from setting the mode of a scenario, the SCADA is responsible for the following fields:

- Forecast: the SCADA should store information in its database regarding energy price, demand forecast and meteorological conditions and forecast;
- Profiles: profile generation and definition, their communication towards each LC;
- Scenario: The SCADA gives all the preparation and parameters to the LCs to start a scenario emulation:
    - Minimum and maximum active power in W,

  - Minimum and maximum reactive power in var,
  - The type and profile of the emulation,
  - The mode of each LC,
  - The initial minute of the profile when the demo starts,
- Control: the SCADA should be connected to all elements of the network and be able to start, stop, enable or disable any of them. Moreover, as mentioned before, the SCADA shall be responsible for the demo configuration.

The principal role of the other main body, the EMS would be to collect and transmit information from external databases (Influx Database), and to create long-term strategies for control mechanisms. These information packages include historical data of demand, prices, forecasts, and any other types of necessary, previously configured information. These packages are then sent to the SCADA, which stores them in its MySQL database.

2.3.3.2. Monitoring module

The monitoring and supervising module is responsible for the data acquisition, transmission and storage, as well as the real-time control of the microgrid network. Figure 14. shows the communication exchanges between the central bodies.

Under the monitoring module the SCADA is responsible for the following fields:

- Historical Data Treatment: data storage in MySQL Database, presentation of results, historical data, and reports,
- Acquisition of Data: individualized information shall be collected and concentrated from all devices, in real time. For such purpose, the SCADA would be equipped with communication drivers, TCP/IP server, Modbus TCP/IP client, and an MQTT Client Driver.
- Alarms: generation of alarms based on pre-defined normal and not normal conditions of the LCs,
- Real-time data visualization: graphical presentation of the pre-configured parameters, as a function of time,

The EMS would be provided information by the aggregator, as well as by external databases (Influx DB). The EMS sends these information packages to the SCADA via Modbus TCP/IP to be used for control.

The SCADA is connected to the LCs via Modbus TCP/IP and supervises as well as controls each of them. It receives and transmits data of each of their states, and sends them active and reactive power setpoints, which then the LCs apply in their sequence. These setpoints may be calculated by the EMS in case of auto mode, and by the SCADA in case of manual mode.

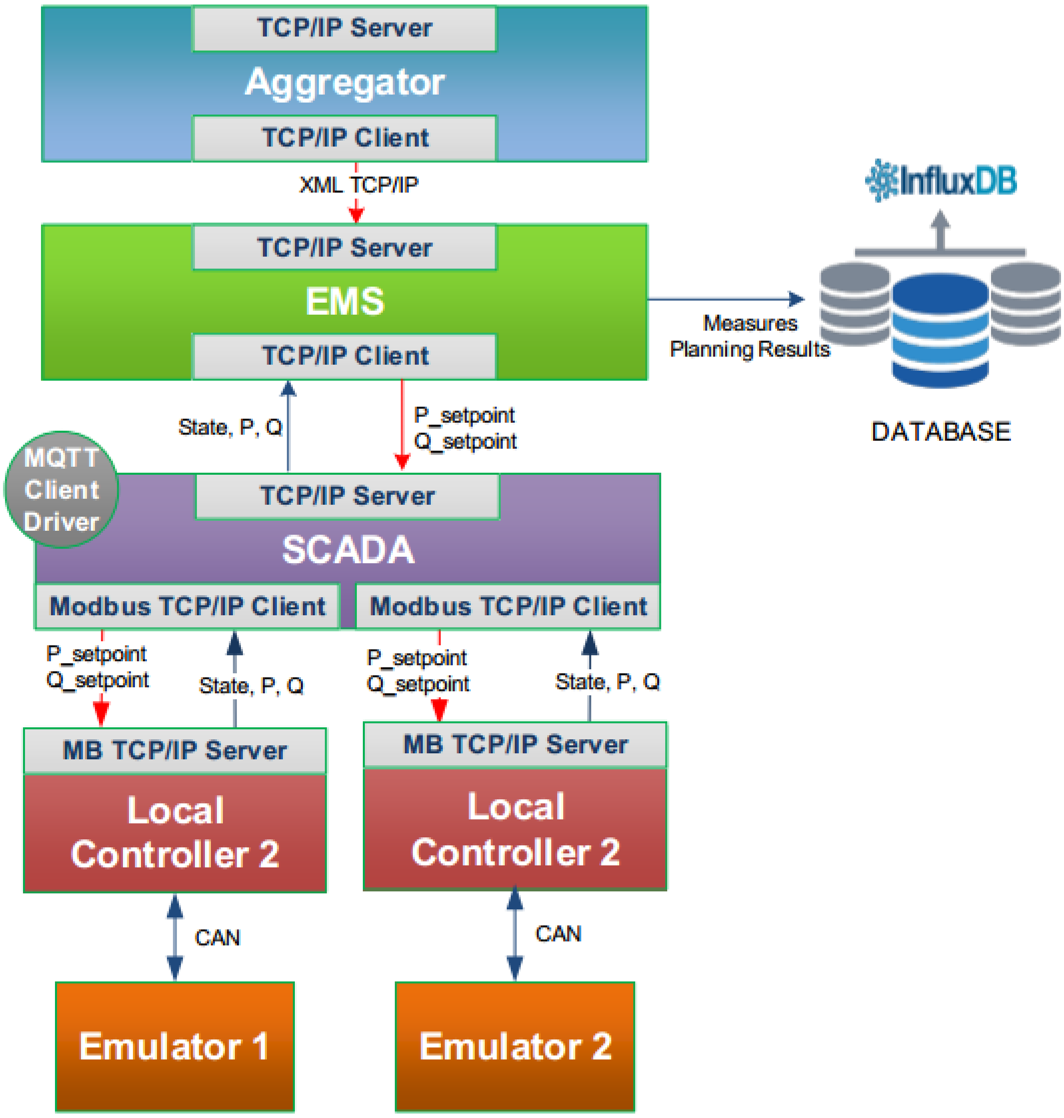


Figure 14: Information exchange between the participants in the monitoring module

3) Development of the microgrid SCADA system

3.1. Functional requirements of the SCADA development

The SCADA has been developed using Schneider Electric's Wonderware InTouch Machine Edition (ITME) program and currently can only be accessed locally in IREC. The program, i.e. ITME project, shall be in charge of centralizing all the laboratory equipment in one place, being monitor, control and operate them from a single computer. Chapter 2.3.3. lists the different functionalities of the SCADA according to the future plans of SmartLab. Nevertheless, as most of the laboratory equipment is currently under development, the developed SCADA focuses on the connection, supervision, control, and data acquisition of the LCs. The following requirements defined the primary guidelines of the system:

- Establishment of connection with three laboratory LCs: RB16, RB04 and MYD1;
- Generate and edit scenarios: the process of generating and editing scenarios should be executed independently from the operation and supervision;
- Start the microgrid elements in manual / automatic mode;
- Stop the microgrid cabinets;
- Store historical data of set and real active power;
- Possibility of creating alarm panels with the record of incidents;

3.2. Introduction to ITME Development Environment

The following Chapter discusses the main particular tools and vocabulary of the ITME development environment.

ITME projects generally have several screens, which are windows on which the Human-Machine Interface (HMI) is executed. Screens can have several sizes, and a number of them can create a screen group. A cluster, which is organized as one, having different screens at each part of it. On each screen there are shapes and objects taking place.
Tags are the basic variables of an ITME project, which store data obtained from communication with communication devices, from the results of calculations or from user input.

They can be booleans, integers, real numbers, or strings, and may be multidimensional. Moreover, tags can refer to a class, which are customized structures of data, i.e. class members. These members may also behave like tags under certain conditions.
Tags are usually linked with one or more address of a register of a connected device. The connection is facilitated by a communication driver, a pack of small programs which enables the functioning of a communication protocol, as well as it contains information about the device to connect with. During driver configuration, the ITME project gets linked with a remote device, e.g. with a LC. For driver configuration in ITME there are two possibilities: Main Driver Sheet and Standard Driver Sheet. The first is suitable for low linked tag number, but high connected address number. The latter one is more applicable for applicable for situations, at which there are numerous tags to get linked to a single port.
Historical data storage is important for the operators in order to revise and analyse the trends and previous values of project variables. ITME trends facilitate linking the trend curves of project tags with external databases. For the graphical representation, recovery of history data, ITME trend control object may be used. Figure 23 is an example for such representation.
Alarms in ITME are created in order to notify the operator about any problems or abnormal conditions. Alarms and their text messages are linked to up to 4 values: high, low, and very high, very low values of any tags.
Lastly, ITME provides the possibility to customize a project by writing scripts in VBScript. Scripts can be written in different parts of an ITME project, and they may be executed as background tasks, as well as supplementary tools for the design.
Regarding security and permissions of a SCADA project developed in ITME, the program gives the opportunity for the creation of groups and users. Each user has a password to access the system and usually they belong to a person. A group is a group of users with the same specific authorization to access or to send information in the SCADA. Each group admits its users and it is the group that determines the permissions of the users. Each user would have their own credentials to access the system even if they belong to the same group.

For more details about ITME tools used for the development, please refer to Schneider Electric’s ITME user manuals. [45]

3.3. Data acquisition and management

The SCADA must unify the different elements and connect with all of them. During driver configuration, the ITME project gets linked with the LCs with the help of the built in Modbus TCP/IP driver. As one LC contains numerous pieces of information of the same type, a Standard Driver Sheet has been chosen for the configuration. The linked tags and driver configurations are further described in C.

Currently all the available LCs (RB16, RB04, MYD1) are programmed under the conditions of the old configuration. As such, all the connected LCs are operated under port 1500 and are configured under port 1501. (refer to Chapter 2.3.2.2) The configurations have been established according to Table 7.

Table 7: Established connections with the LCs

Name of LC:	IP address:	Port Number	PLC ID:
RB16	192.168.0.116	1500 and 1501	1
RB04	192.168.0.104	1500 and 1501	1
MYD1	192.168.0.101	1500 and 1501	1

Figure 13 and figure 14 show the planned data flow between each participant of the microgrid. To store the data, and to avoid loss of information, a database management system of MySQL has been established. MySQL has the advantage of making it possible to view the data without storing it into the object. Moreover, MySQL uses a long-established standard, being adopted by ANSI, ISO. Amongst others, these are the main reasons why it is used widely in many fields, such as Data Integration Scripts or Analytical Queries, or numerous web applications.

The database has been currently set up on the same computer where the SCADA runs. The ITME project communicates with it to save configured parameters. The SCADA currently uses 1 second as a saving trigger, which could be changed if disk saving is needed. At this point, all active, reactive powers, and the SOC of the battery are being stored with the primary key of writing timestamp. The timestamp has a datetime type, and all other variables are saved as double data type.

Apart from saving information, the ITME also communicates with the database management system to obtain the values and to allow the visualization and management in real time or in historical terms.

3.4. Human-machine interfaces

The following chapter describes the different features and human-machine interfaces in the SCADA. Each section discusses a screen with its main role and structure. This initial part is then followed by the exact description of the features. All the details of the codes programmed are published in annex and are referenced under the corresponding section.

3.4.1. Initial screen

The initial screen of the SCADA can be seen on Figure 15. The screen itself can be distinguished into three parts: the top bar, the navigator pane and the main screen. This latter one is the frame which always changes according to which part of the program we are on. On the top right side there are the IP address of the computer, the date and time, and the exit buttons shown. Moreover, in case later some parts of the system get protected, the user and group settings are also prepared. At this point, the developed system does not have any user-specific right-restrictions, however, later these features may also be added.

Figure 15: Initial screen, screenshot

3.4.2. Configuration screen

The role of the configuration screen (Configuration.SCC) is to enable the user choosing the guiding parameters of the upcoming simulation, based on the configuration module described in Chapter 2.3.3.1. The configuration parameters are stored externally in a .csv file. A screenshot of the configuration screen is shown in Figure 16. The screen can be distinguished to three parts: the part where the file treatment is executed (save and load button), the local controller-specific symbol, and to the background explanatory drawings.

There are 4 scripts called under this screen for the different features:

- configuration file load,
- configuration file save,
- profile configuration general script,
- and profile configuration file change.

All these scripts can be further studied in Appendix G. The exact places at which these scripts are called are referenced in the corresponding part below.

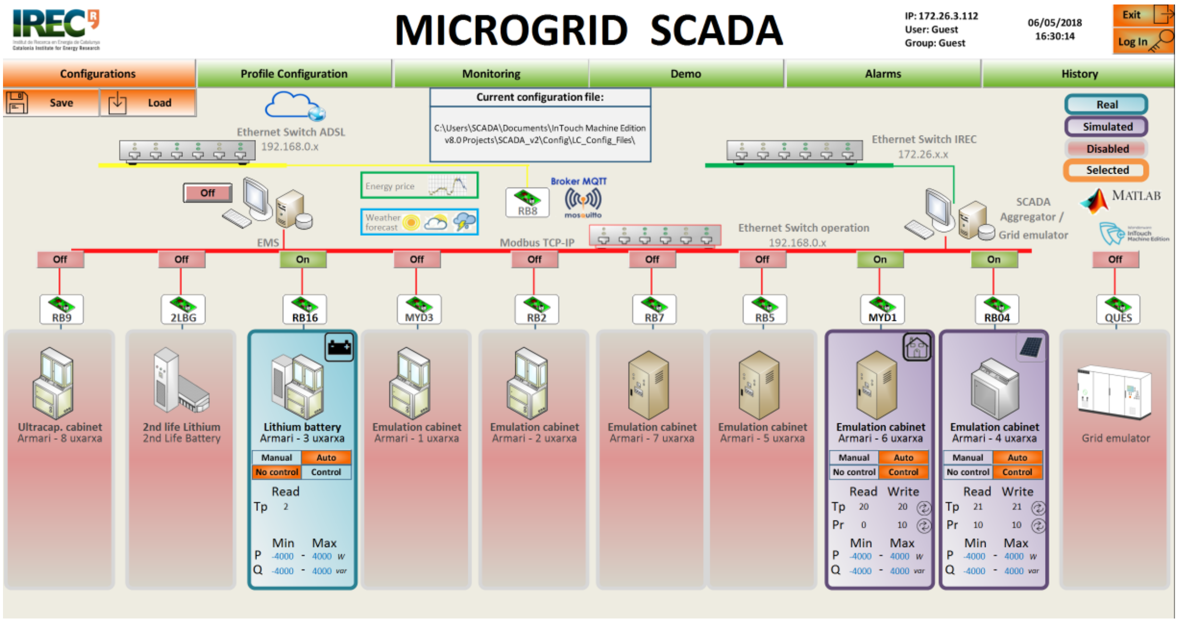


Figure 16: Configuration screen, screenshot

3.4.2.1. Load and save buttons

The load and save buttons of the configuration file are placed on the top-left corner of the screen. Clicking on the save button, the user has the possibility to save the current configuration data of all local controllers configured in the SCADA system in a chosen or newly created .csv file. This way the configuration data of the SCADA can be changed, saved and a previous configuration setup can be loaded in an easy-to-use environment. These files are located under the project folder, *Config\LC_Config_Files*. There have been two scripts written for the realization of loading and saving .csv files: the script of configuration file save, and the script of configuration file load. For more information about the exact codes, animations and commands written, please refer to Appendix H.

3.4.2.2. The local controller-specific configuration symbol

There are nine symbols located in the middle of the screen, each of them may have a red X if the LC is enabled but connection is failing to the SCADA system. Figure 17 shows the developed symbol, used as a representation of all LCs. The symbol is made in a parametrized way, so that when anything is changed on it, all the 9 LC-representing figures change. Each symbol gives the possibility to the user to configure the following parameters:

- Enabling or disabling the LC: this feature on the upper-edge of the symbol helps the user selecting the systems with which the upcoming emulation shall run.

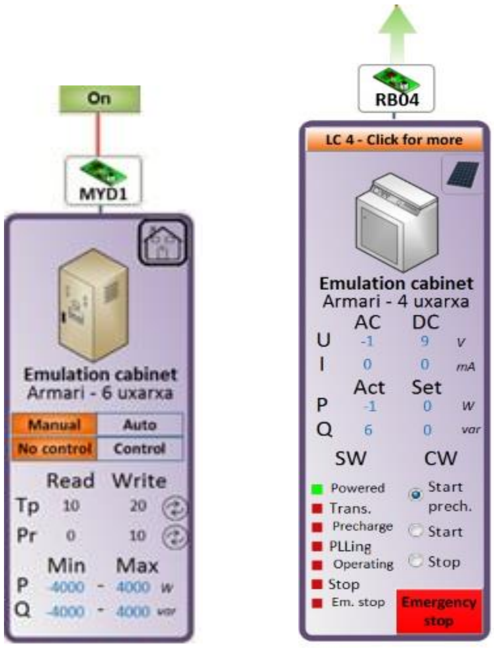


Figure 17: The configuration (left) and the monitoring (right) symbol

- Change the mode of the LC: the mode of a LC can be either manual or auto and can be changed by the user at any point. The significance and sense of these modes are described in Chapter 2.3.2.

- Change the controllability of the device: the user can change the controllability of the emulator, if applicable. The significance and sense of controllability is further elaborated in Chapter 2.3.2.
- Type selection: The importance of a selected type is explained in Chapter 2.3.2. In case of real devices, the user cannot change the type, only the read data is presented. However, for an emulation cabinet the type change can be done via clicking the refresh sign on the right side of the symbol. This action opens a popup window (Configurations_TypeChange.SCC) which facilitates the user to select between the available emulated types. On Figure 18 it can be seen that an orange frame indicates the selected local controller. The list of available types may be modified by changes in the in the start-up script configuration. In case there is a new type selected, the profile number automatically switches to the first .csv file of the corresponding folder. The user can exit this popup by either clicking on OK or Cancel. There have been two scripts used for the realization of type selection: profile configuration general script, and profile configuration file change. After selecting the desired type, the code of that type is written in the column of written type data (*Type*), which overwrites the Modbus TCP/IP address (*RW_Type*) when configured. For more information regarding the

solutions used in ITME for this screen, please refer to Appendix H.

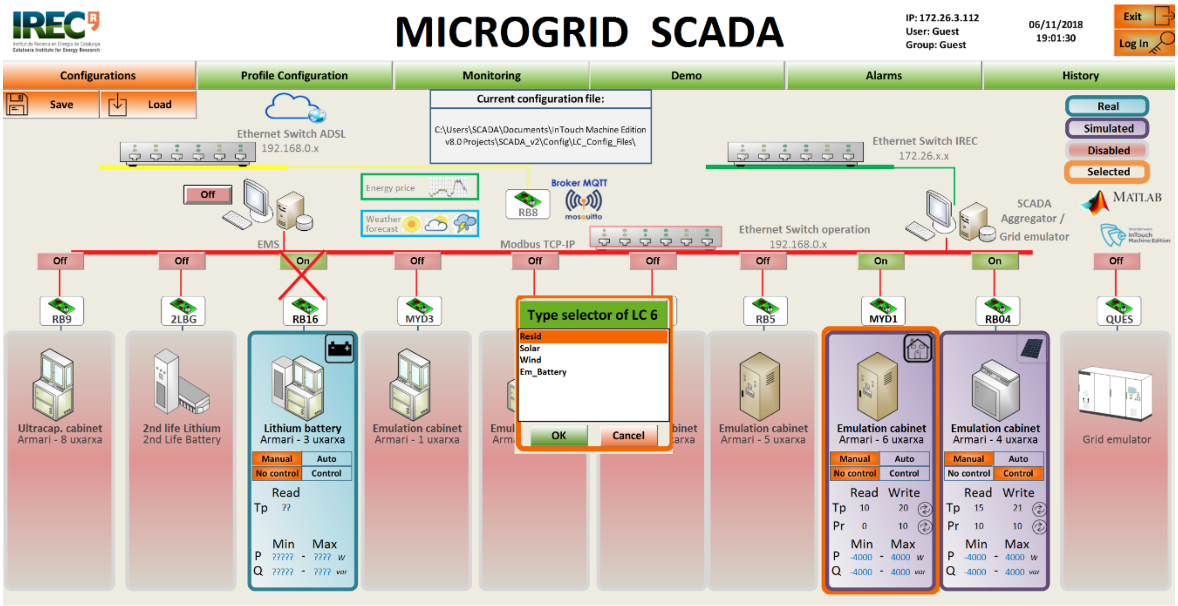


Figure 18: Type selector popup on the configuration screen, screenshot

- <u>Profile selection:</u> The importance of a loaded profile is explained in Chapter 2.3.2. In case of real devices, the user cannot change the profile, it can be only the operating conditions always follow the control sings, hence, only the read data of profile is shown. However, in case of emulated devices the user may change the profile of the device. It can be done via clicking the refresh sign on the right side of line. This action opens a popup window (Configurations_ProfileChange.SCC) which facilitates the user to select between the available .csv files of the corresponding chosen type. On Figure 19 it can be seen that an orange frame indicates the selected local controller. The list of available profiles may differ by changes in the corresponding folder. Moreover, on the right side of the popup the selected profile's graph can be seen. The user can exit this popup by either clicking on OK or Cancel. There have been two scripts used for the realization of profile selection: profile configuration general script, and profile configuration file change. After selecting the desired type, the code of that type is written in type write (*ProfileNum*), which overwrites the Modbus TCP/IP address (*RW_ProfileNum*) when configured. For more information regarding the solutions used in ITME for this screen, please refer to Appendix H.

- <u>The minimum, maximum active and reactive power:</u> In the symbol these data are directly written on Modbus TCP/IP. The importance of these parameters is presented in Chapter 2.3.2. For more information about the exact solutions used in ITME for having the upper presented features, please refer to Appendix H.

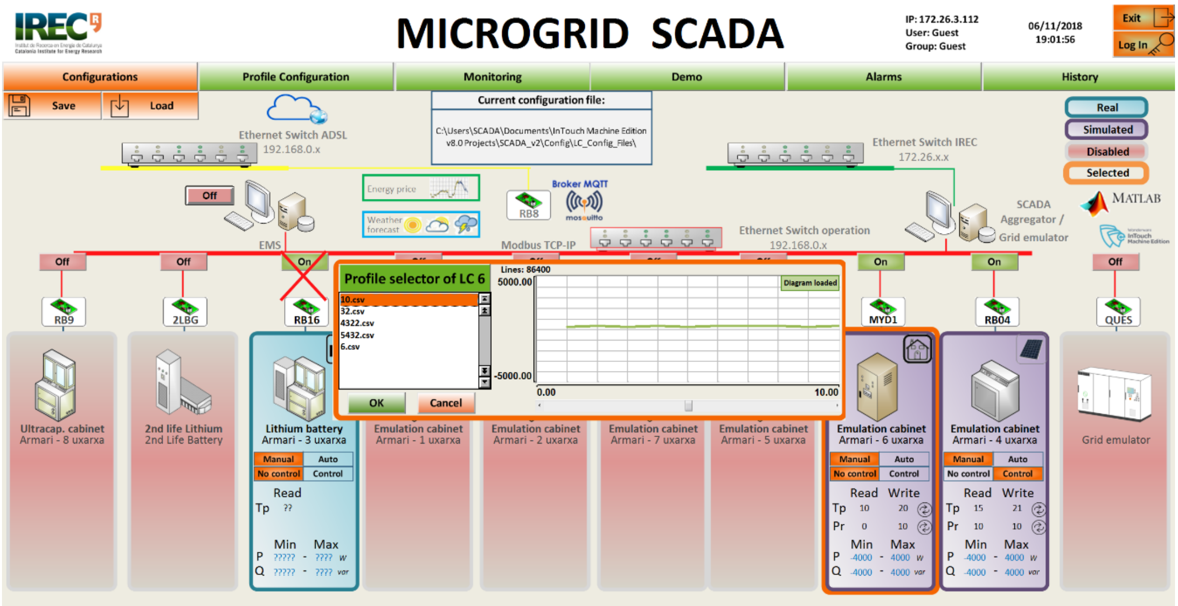


Figure 19: Profile selector popup on the configuration screen, screenshot

3.4.3. Profile configuration screen

The role of the profile configuration screen (Profile_Configuration.SCC) is to enable the user checking and then editing the available profiles of each configured type. The profile files are stored in .csv format under *Database\Profiles* in a folder corresponding to the type description. As it can be seen on Figure 20, the screen can be grouped to two parts based on their functions: on the left side the user can change what they would like to configure, and then see the resulting graph on the right side. A specific type can be selected in the list box above. After changing the selected type, the corresponding files appear in the list box below. There have been two scripts used for the realization of this feature: profile configuration general script, and profile configuration file change. Afterwards, the user is given the opportunity to configure and to open the profile file in Excel directly from the SCADA. For more information about the exact codes and commands written, please refer to Appendix H.

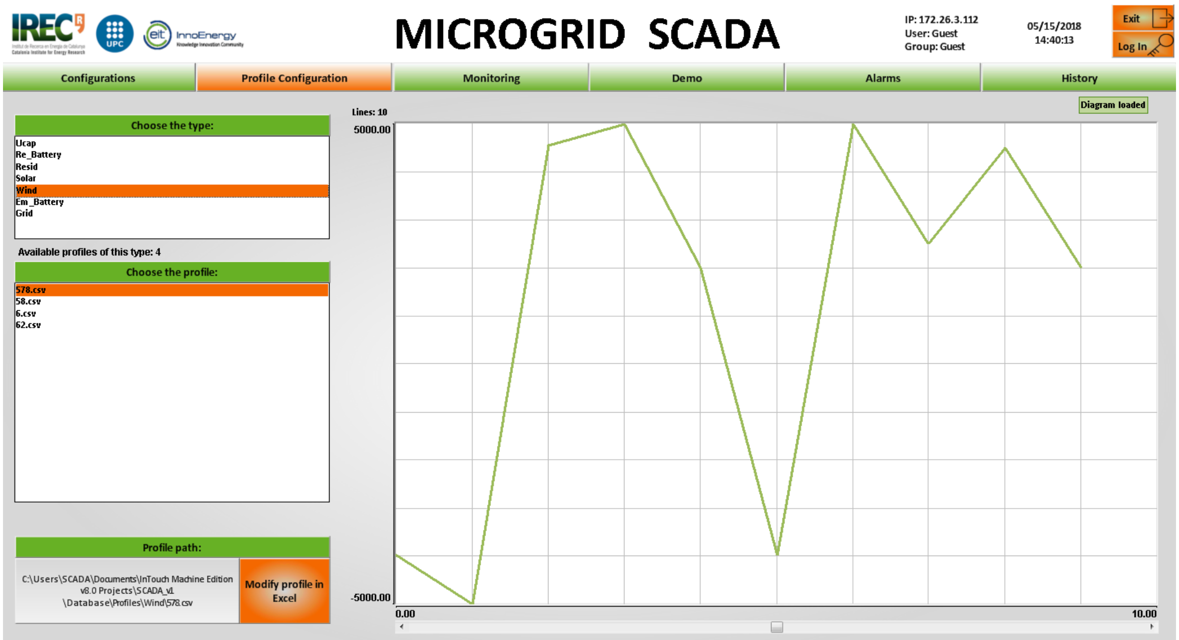


Figure 20: Profile configuration screen, screenshot

3.4.4. Monitoring screen

The role of the monitoring screen (Monitoring.SCC) is to enable the user monitoring and to enable sending some particular messages to each LC. The functioning of the monitoring screen is based on the monitoring module described in Chapter 2.3.3.2. The screenshot of this screen can be seen on Figure 21. It can be distinguished to two parts: the local controller-specific monitoring symbols, and to the background explanatory drawings. Moreover, a popup screen of detailed monitoring information is available from this panel.

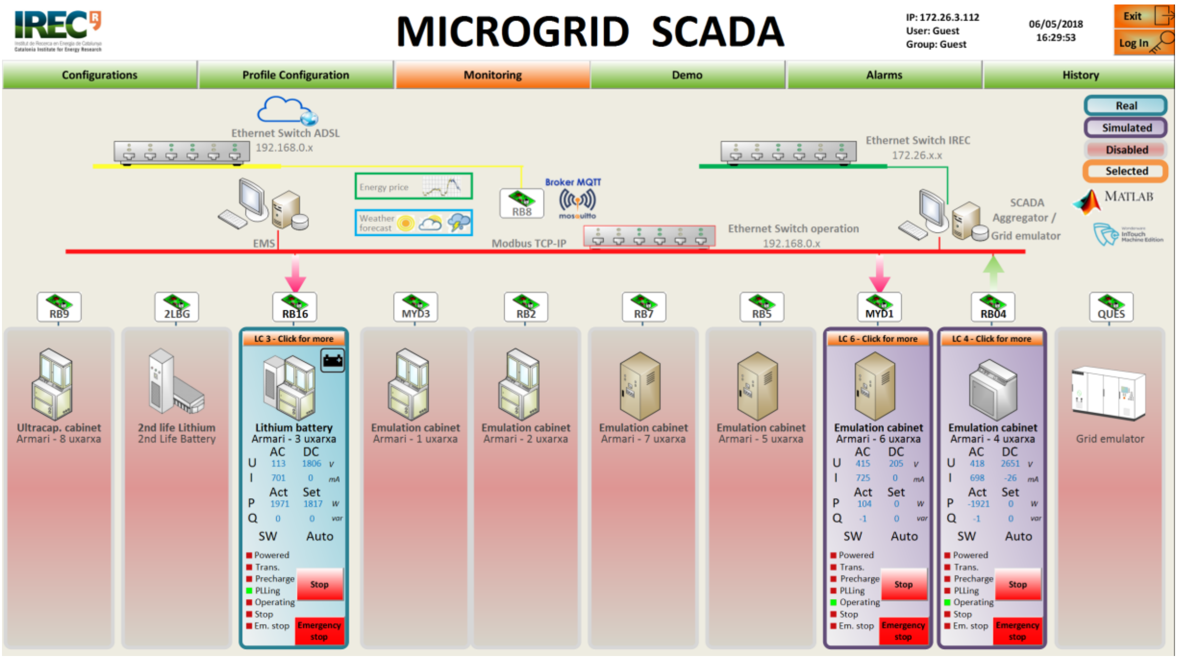


Figure 21: Monitoring screen, screenshot

3.4.4.1. The local controller-specific monitoring symbol

There are nine symbols located in the middle of the monitoring screen. Each of them may have a red X if the LC is enabled but not connected to the SCADA system. Figure 17 shows the developed symbol, used as a representation of each LC. The symbol is made in a parametrized way, so that when anything is changed on it, it changes on all 9 LC-representing symbols. There is a button on the top of each symbol which facilitates reaching all the available information of the LC. On the symbol itself the following parameters can be seen:

- Arrows show the direction of active power flow,
- In case of any error, the error message arriving on Modbus TCP/IP (see Chapter 2.3.2.3.) appears above the name of the LC,
- The AC, and DC current and voltage in V and in mA,
- Set and actual, active and reactive powers in W and in var,
- The actual SW of the LC.

Moreover, to facilitate controlling all the LCs from one computer, the possible CW messages in case of manual, and the start/stop buttons in case of auto mode are shown. In any case the user is given a button for the emergency stop of the equipment. For more information about the exact codes, animations and commands written, please refer to H.

3.4.4.2. Detailed information popup

The detailed monitoring screen of a LC (Monitoring_More.SCC) shows all the available, real-time information of the LC in a format of table. On Figure 22 it can be seen that an orange frame indicates the selected LC. The detailed information shows all the real-time data available about the LC, according to the Modbus TCP/IP mapping described in Chapter 2.3.2.3. For more information about the Modbus TCP/IP variables and of the screen configuration, please refer to Appendix C and to Appendix H.

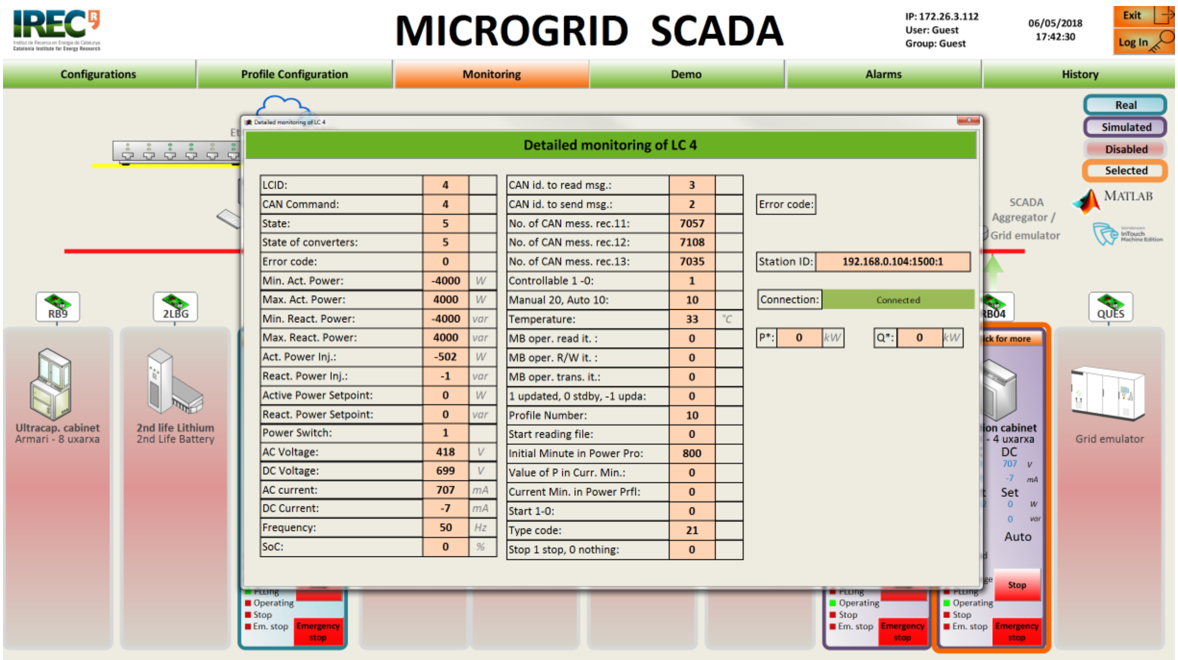


Figure 22: Monitoring details screen, screenshot

3.4.5. Alarms screen

The role of the alarms screen (Alarms.SCC) is to enable the user reviewing any configured event in the microgrid. The screen has primarily one part: the list of events in table form. Its screenshot can be seen on Figure 23. Currently, disconnection and error of any LC are listed as alarms. Nevertheless, throughout the operation of the microgrid, further elements may be added for any condition out of normality.

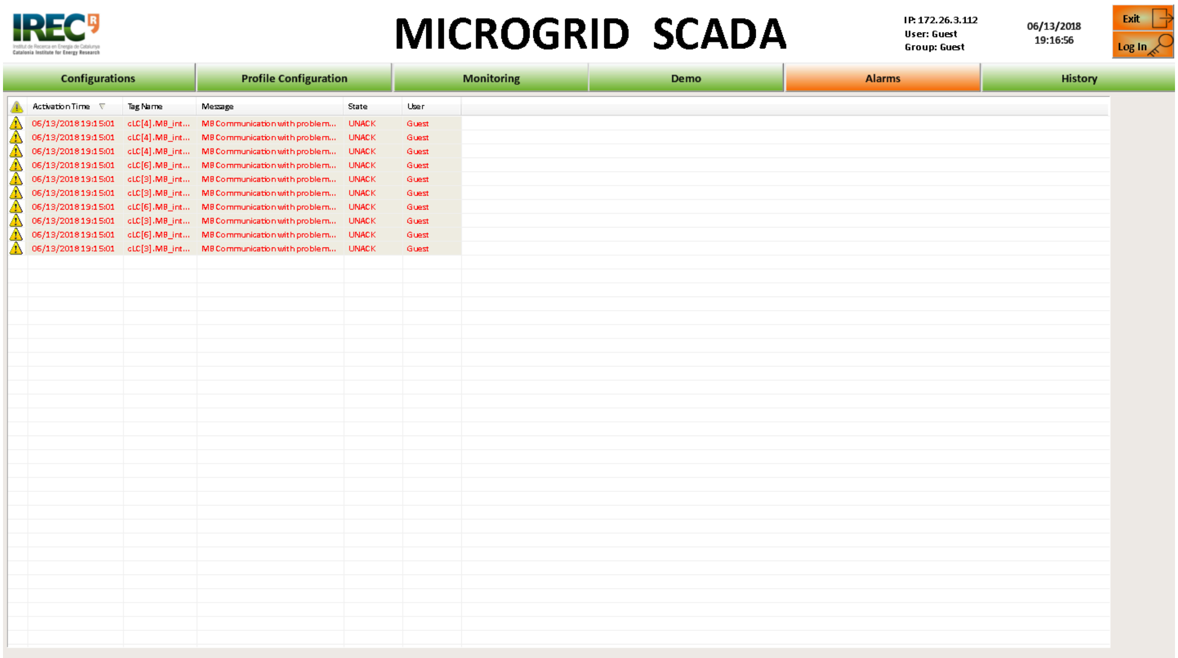


Figure 23: Monitoring details screen, screenshot

3.4.6. History screen

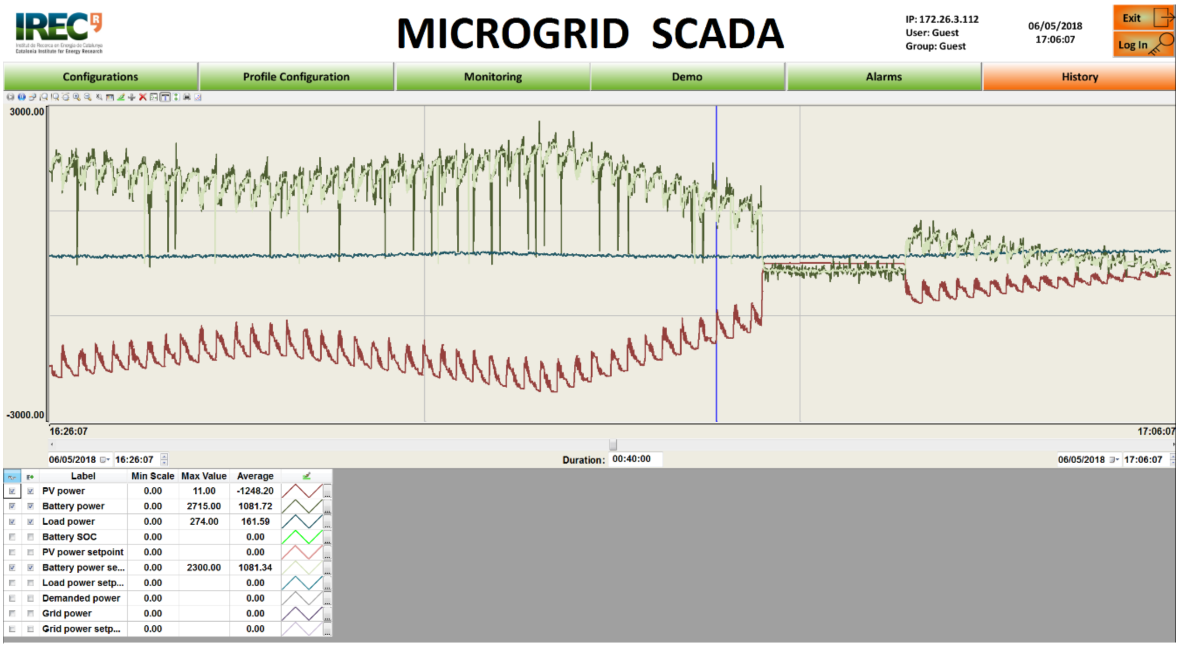


Figure 24: History screen, screenshot

The role of the history screen (History.SCC) is to enable the user reviewing any configured data of the LC of any saved time period before. The screen has primarily one part: the chart of data which can be configured in several ways. Its screenshot can be seen on Figure 24.

The conditions of data treatment are described in Chapter 3.3. All the configured data are stored real-time in a MySQL database. From that, selected range of data is shown here in the chart. The database is accessed through ADO.Net technology in ITME. As the time save and access are continuous, the screen is appropriate for analysing current trends. For more information about the screen configuration, please refer to Appendix H.

3.4.7. Demo screen

For each project done with this SCADA and the microgrid, it is expected to create a different demo screen, on which the project-specific environment is developed. The role of the developed demo screen (Demo.SCC) is to enable the user executing the showcase experiment with the participation of three, previously configured cabinets: the PV, and the residential load emulation cabinets, and the SAFT battery. This showcase experiment of the microgrid laboratory SCADA system is further described in Chapter 4. The screenshot of the showcase demo screen can be seen on Figure 25. The screen can be distinguished into three parts: the demo board, the general monitoring of the emulation, and the history curves.

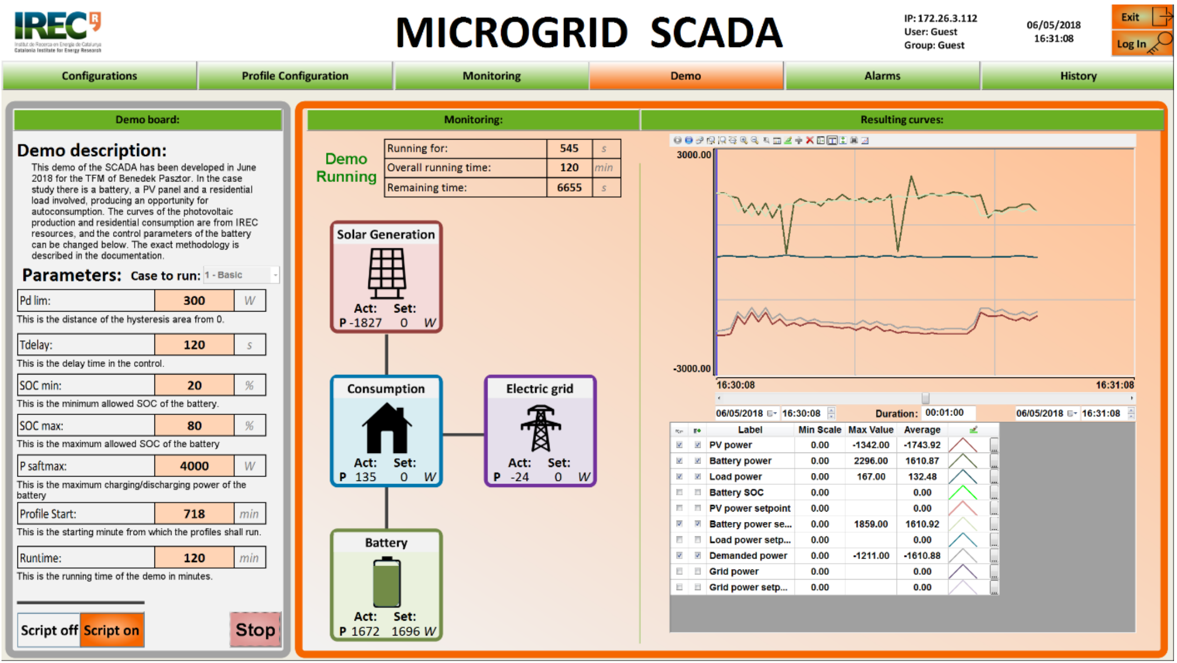


Figure 25: Demo screen, screenshot

The demo board enables the user getting general information about the showcase, as well as choosing the required parameters of the control developed for the battery. The control, and the importance of the parameters are described in Chapter 4. On the bottom part of the board the user can turn on or off the demo-guiding state machine algorithm, described in Chapter 3.5.2.

In the middle part of the screen the user can monitor the set and the real values of the participating elements of the demo. On the top, information about the demo running time can be read. On the right part of the screen historical curves are available for the previously defined values: the measured power of the PV, the battery, and of the load, the set power and the SOC of the battery, and the calculated set and real power of the grid.

3.5. State machine algorithms in the SCADA

The most important part of communication between the SCADA and each LC is via the state machine algorithm. There have been two state machine algorithms developed in SCADA: one general for the newly configured LCs, and one showcase-specific algorithm for the demo.

3.5.1. State machine for the newly developed LCs

The state machine algorithm from the newly configured LC point of view has been analysed in Chapter 2.3.2.4, and here the planned SCADA point of view is presented. The algorithm is developed for the newly implemented LC configuration; however, some parts are also important in case of the old one. These are indicated in Figure 26 by different rectangles.

Depending on the actual SW there are several actions to get executed in the program. On the configuration screen the user can choose between manual and auto mode. The importance of manual and auto mode has been described in Chapter 2.3.2. In case of manual mode, all actions are supervised by the user, hence, it shall only be used by advanced users who have the exact knowledge regarding the commands (CWs) to send at each situation. In case of auto mode, the cabinet automatically precharges. Having the emulation cabinet in SW="Precharge" (3) means no particular actions from the SCADA point of view, only the status itself is indicated. When the LC is in SW="PLLing" (4), the system checks if the profile number on SCADA (*ProfileNum* property of the LC). If it is, and the read Modbus TCP/IP data of "Profile actualized" is 0, then the SCADA toggles this variable, and timestamps the time of this toggle. Then, it redoes the same process. If the actualization command did not have any effect on the value of RW_ProfileNum in 5 seconds (changeable variable by Timeout_profile), then the system gives an error of profile timeout. When the two profiles are the same, then the button for MB Start is enabled, eventually toggling MB Start=1 when clicking. It puts the LC in SW="Switching" (5). When the LC is in this status, the SCADA would check if the particular LC is controllable or not. This part of the state machine may differ demo to demo. If yes, then it starts to control Modbus_P according to the previously programmed control. If not, then the sequence restarts, and in any case, the "stop" button gets available for the user, which might toggle MB Stop at any moment of clicking. When the system gets stopped, the SCAD shows a green light for SW="Stop" (6). Throughout the whole sequence the "Emergency stop" is available for the user, and in case SW="Emergency stop" (7), it automatically disappears.

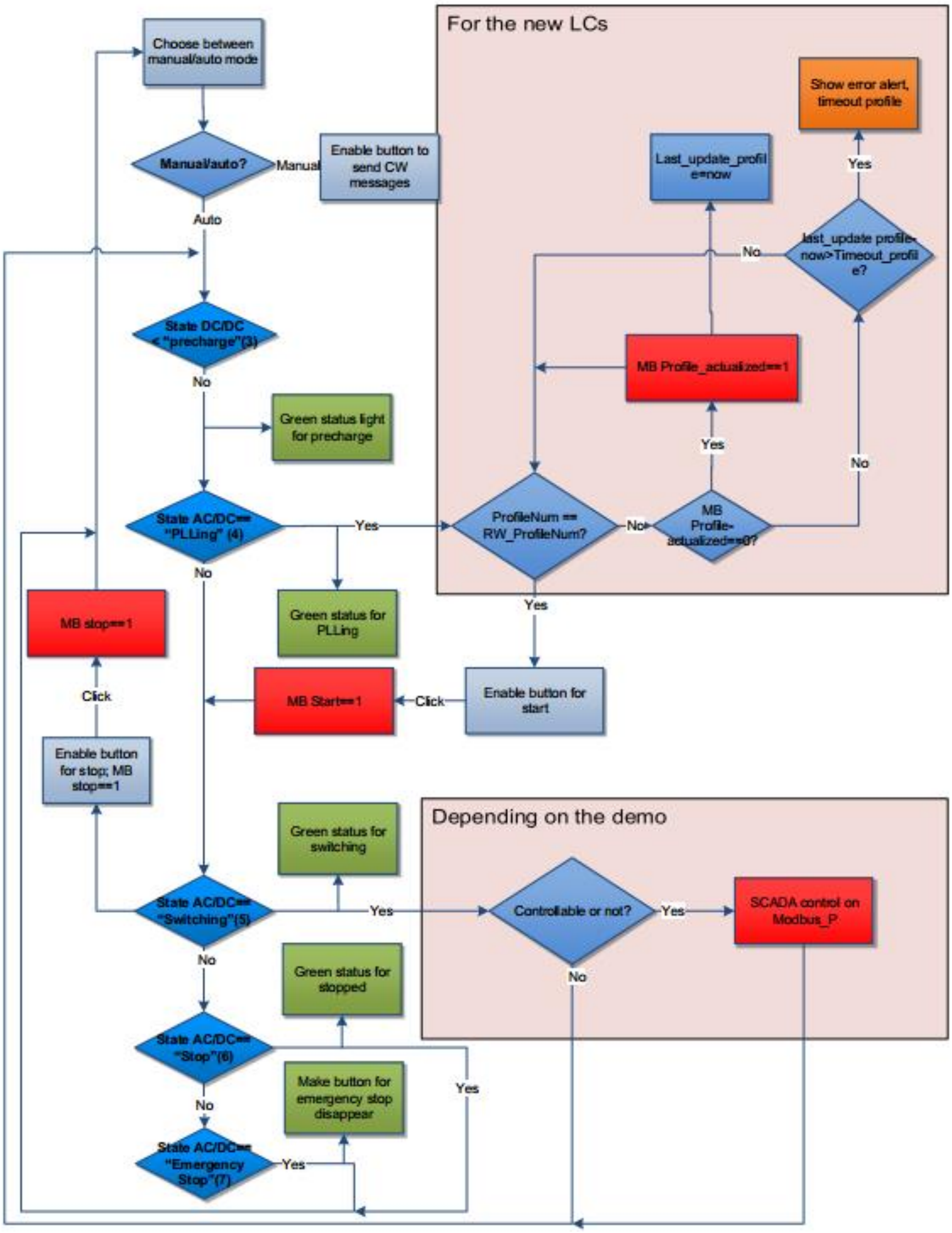


Figure 26: General State Machine algorithm of the SCADA

3.5.2. State machine algorithm for the showcase demo

There has been an executing state machine algorithm developed for the showcase demo. The role of this algorithm is to let the user run the three, old types of LCs smoothly together. The actions listed are required in a particular order for the old LCs, as it is described in Chapter 2.3.2.2. Figure 27 shows the flowchart of the algorithm, which has 5 states:

- State="Standby" (-1), which enables the prepare demo on the screen. During this state the SCADA sends a type 0 to all three elements. When all LCs received 0, and are in SW="Operate", then the user may click on the prepare button, and the program switches to the next step.
- State="Prepare" (0), which enables the demo start button on the demo screen. During this state the profile number, profile start minute and the type of each LCs. Moreover, the CAN properties are set. When the user clicks on start button, the algorithm switches to the next state.
- State="Starting" (1), which puts all cabinets in auto mode and all start switching. These commands are followed by toggling of a tag (InitialValues), whose importance is explained in Chapter 4.3.2. Afterwards, the script switches to the next state, and creates a timestamp (Start_TimeStamp). After conducting all these, the program switches to the next state.
- State="Running" (2), which toggles the control algorithm each second. The configured runtime is compared to the running time, and when they gets equal, the next algorithm switches to the next state. This switch may happen manually as well, in case the user clicks the Demo stop button on the screen.
- State="Stop" (3), which stops the battery-control algorithm, switches the timestamp to 0, and switches all participating LCs to manual mode. This command is followed by the switch to state="Standby" (-1).

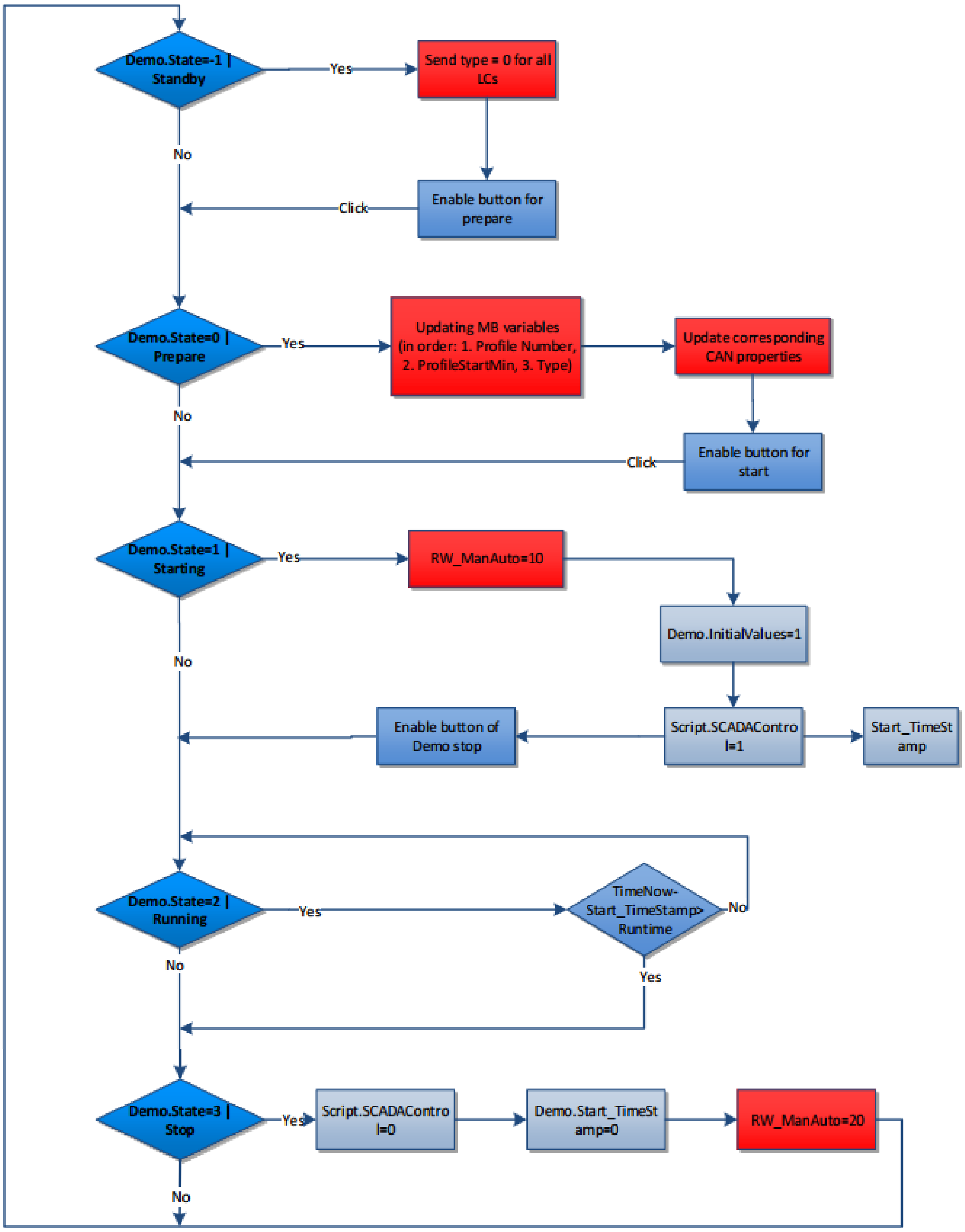


Figure 27: State Machine algorithm for the showcase demo

4) Showcase of the developed microgrid SCADA system: autoconsumption of a prosumer

4.1. Goals and boundaries

In order to test the developed SCADA system, there has been a showcase experiment run on the system with the participation of 3 elements. The setup has been modelled in accordance with Chapter 3.4.7, Demo screen. A residential grid-connected house has been modelled with a photovoltaic panel on its rooftop, as well as a battery in the basement.

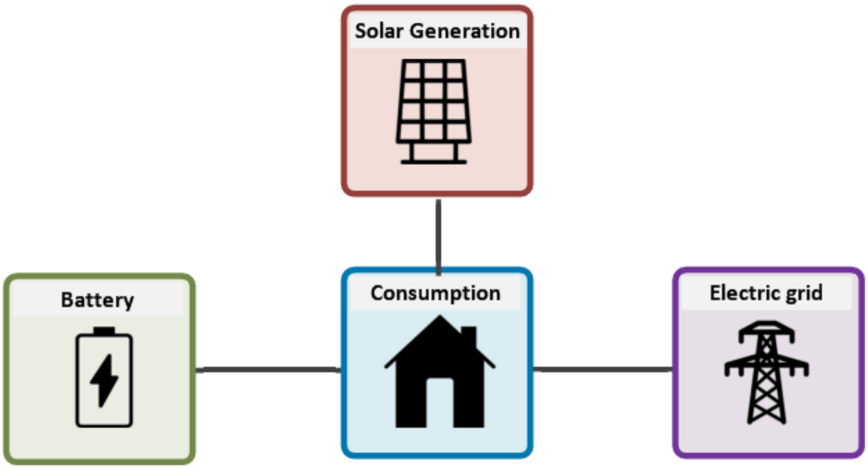


Figure 28: The showcase setup as seen in the SCADA

The ultimate goal of this experimental setup has been to test the developed SCADA, as well as to run an experiment regarding the autoconsumption of a prosumer: a domestic consumer which produces electricity on its own. The PV generation and the residential consumption curves have been provided by IREC from previous works. Two types of simple control techniques of the battery have been developed and compared. Further goal of the showcase is to underline the differences between an emulation and a simulation by conducting the experiment both in the laboratory and by simulating it in Matlab 2014a environment.

Considering the upper-mentioned goals, the case study has several neglections with the following boundaries:

- The input data has been given for a day, out of which the emulation has been conducted a two-hour period. The granularity of the showcase is one second.

- Reactive power flow and power losses are neglected.
- The input data is given about the PV and consumption profile at each second. The battery is being controlled by the emulation / simulation manager.
- Maximum rated PV power ($P_{PV,max}$) is 4000 W as well as the battery ($P_{saftmax}$) is capable of 4000 W.
- The minimum and the maximum allowed battery SOC are the 20 and 80%, gotten from the literature. [46]
- As it is a showcase of the laboratory SCADA system, economic optimization has not been taken into consideration throughout the development.

4.2. Showcase description

4.2.1. Initial data and analysis

The power curves of the PV and of the residential consumption have been given by IREC from previous works. Figure 29 shows the received input curves. Negative powers indicate generation, and positive values stand for consumption of energy.

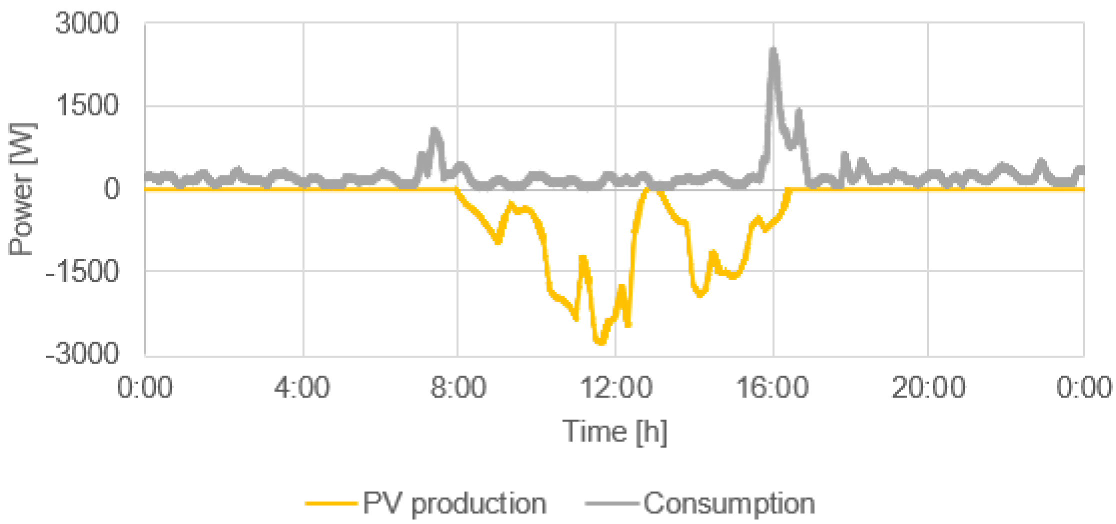


Figure 29: The initial 24-hours long data given by IREC

Analysing the PV production curve, the Sun rises at round 8 am. The production peaks at

2.775 W and has an average of around 380 W during the whole day, 1.068 W during daylight. The overall produced energy is calculated to be 9.120 Wh. There are several drops in the curve during the day, which indicates semi-cloudy weather. The biggest cloud arrives at around 1 pm, and the generation completely stops for around 20 minutes.

The overall consumed energy is lower than the produced, around 6 000 Wh with an average power of around 260 W. This low consumption indicates that the data could come from a small house, possibly a cottage with little needs. Moreover, there seems to be little activity in the house which would require electricity. There is a constant, periodically more or less repeating consumption of around 200-300 W at night, which could be a household device, such as a small fridge. This fridge periodically turns on, which gives the base consumption. The morning routine of the people living inside must give the power increase at around 7:30 am. In the afternoon, at around 4 pm, there is a peak in the consumption. This is probably because of the activity of the people after arriving from work. This peak is followed by the base consumption, meaning little, almost no activity in the house.

Analysing further days of consumption and production could indicate more details about the consumer behaviour, as well as could give more realistic data of the actual autoconsumption efficiency. However, as mentioned in the boundaries, the goal of this emulation is not to optimize the autoconsumption, but to test the SCADA and the laboratory equipment.

4.2.2. Selection of the emulation period

There has been a period of two hours selected from the initial data to work with. As the most interesting part of the day is the arrival of a cloud at around 1 pm, The period of 12 pm – 2 pm has been chosen for the try-out. The exact power curves of these two hours can be seen on Figure 30

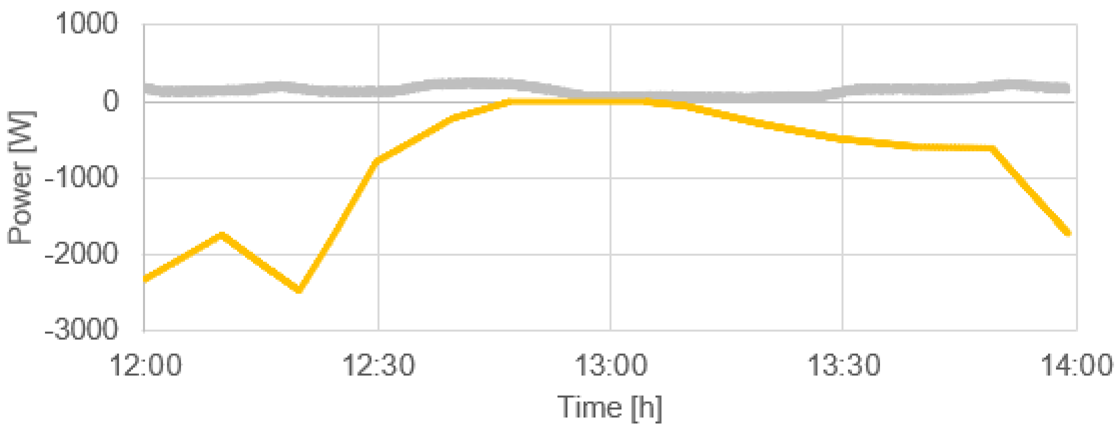


Figure 30: The selected two hours of power profiles

4.3. Simple battery control techniques for autoconsumption

There have been two cases of real-time control of the battery emulated and simulated. For all cases the goal of the control is to set the battery power (P_{saft} in W) for each second based on the following data:

- P_{PV} (photovoltaic generation power) real-time value in W
- $P_{cons.}$(consumption power) real-time value in W
- $P_{saft,\ max.}$(maximum charging/discharging power of the battery) battery constant in W
- SOC (State of Charge of the battery) real-time value in %
- SOC_{min} and SOC_{max} (configured minimum and maximum SOC of the battery) in %

Moreover, the demanded power, P_d, gives the amount of overproduction or extra energy needed at each second. It is defined as the sum of the real-time consumption and generation power:

$$P_d = -(P_{cons} + P_{PV})$$

4.3.1. Control Case 1 – base control

The most basic battery control methods are simply proportional with some limiting conditions. Control Case 1 is simply based on the idea of when there is an overproduction of photovoltaic generation, then the battery is getting charged, and in case of extra need to fulfil the consumption, the battery provides that power and is discharging. This basic control is visualized in Figure 31, showing the battery power setpoint P_{saft}* as a function of P_d.

This control always sets the battery power to covering P_d, unless SOC or the demanded power are out of what is allowed. When the battery SOC is below the minimum (SOC_{min}) in case of discharging mode or over the maximum (SOC_{max}) configured level in case of charging mode, then the grid is completely used for covering P_d. If SOC is in between these two values, then P_{saft}* is controlled to cover P_d until a battery constant $P_{saftmax}$. $P_{saftmax}$ is the maximum allowed charging or discharging power of the battery. Over this power the grid equilibrates the rest of the power needed. The corresponding control flow diagram of Control Case 1 can be seen on Figure 32 and the corresponding codes can be further studied under Appendix G and I.

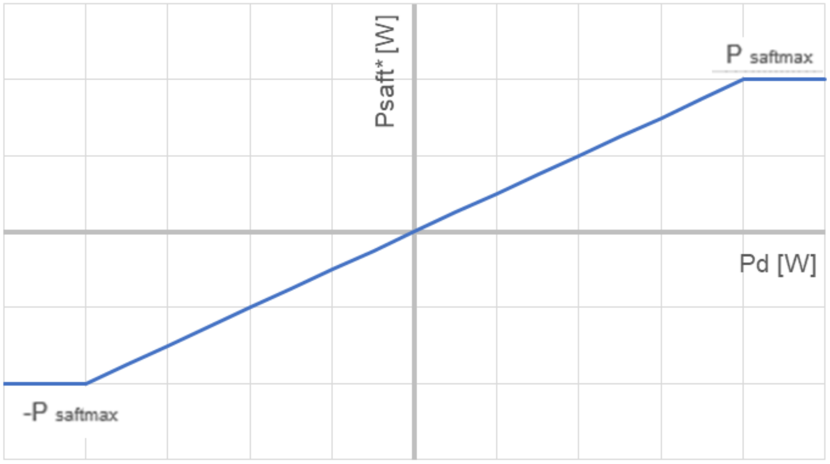


Figure 31: Graphical representation of Control Case 1

The advantage of Control Case 1 is its simplicity. However, there are major drawbacks expected due to mainly its real-time value control type. Namely, sudden and frequent changes of charging/discharging mode of the controlled battery power, which would eventually harm the device. Moreover, Communication delays, and sudden weather changes could cause unstable battery control conditions. To cope with these expected problems, Control Case 2 has been developed.

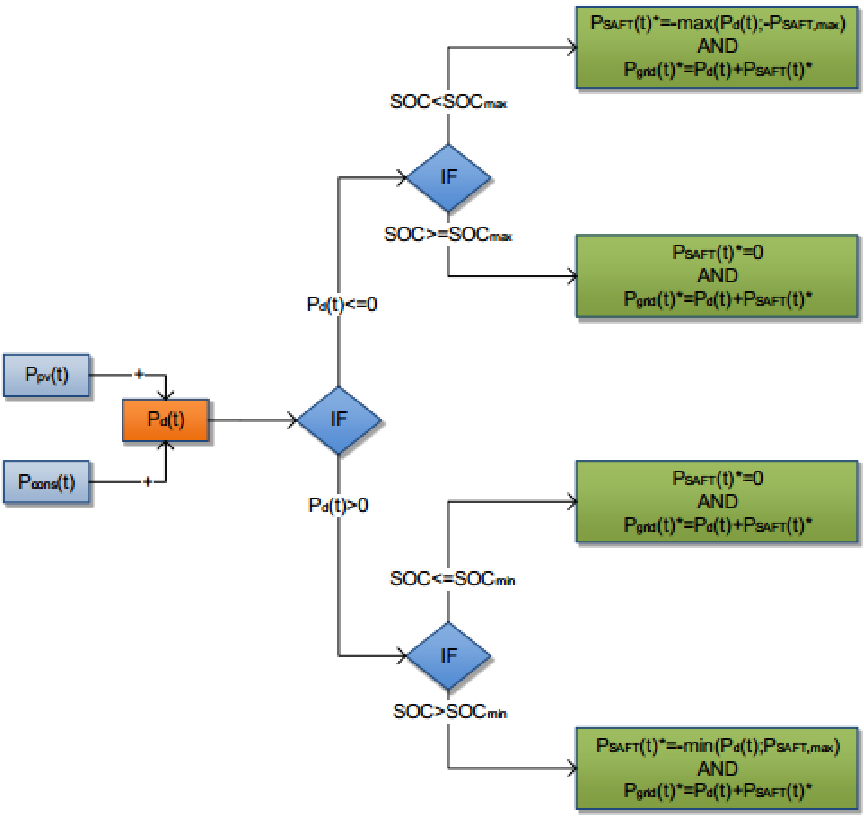


Figure 32:Control flowchart of Control Case 1

4.3.2. Control Case 2 - battery control based on hysteresis theorem

The second idea for the battery control has been based on hysteresis theorem. Hysteresis is a phenomenon in which the value of a property is not only based on the state, but also on the historic circumstances. This phenomenon can many times be observed in nature, e.g. magnetic moment, as well as in the technology, e.g. control techniques. [47]

The control curve of P_{saft}* consists of two separated curves: the increasing and the decreasing curves. As it can be seen on Figure 33, when P_d<0, the increasing curve (blue) has the same behaviour as the control curve of Control Case 1. In the range of P_d>0, the increasing curve keeps P_{saft}* 0. On the other hand, the decreasing (red) curve behaves like the positive half of the curve of Control Case 1. Under 0 this second curve gives a set value of 0.

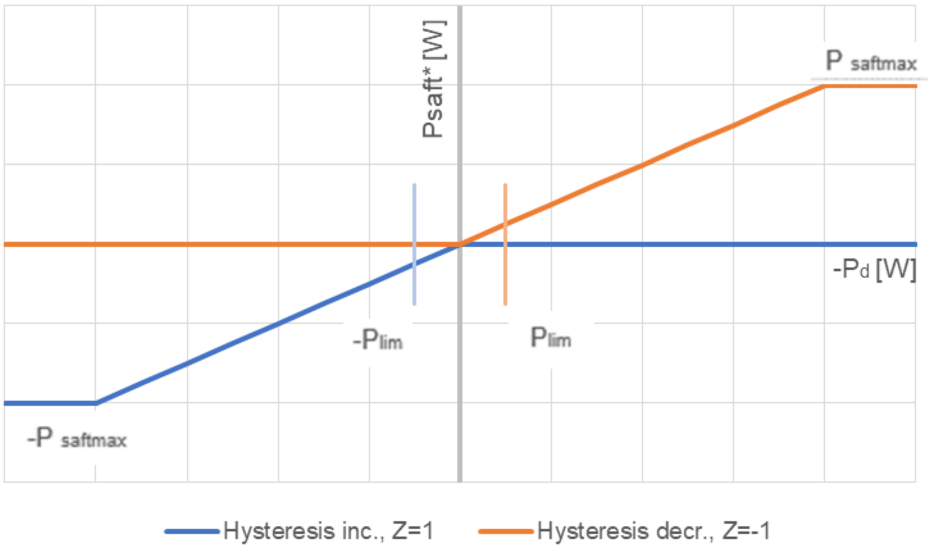


Figure 33: Graphical representation of Control Case 2

The change from one curve to another is only possible when P_d is out of a pre-defined dead band range of P_{lim} around 0 for a period of time of T_{delay}. This control technique based on hysteresis with T_{delay} delayed change aims to mitigate the problems of sudden weather changes combined with a delay of communication.

In case of Control Case 2, the grid is expected to be used more heavily when sudden changes occur in the equilibrium of generation and consumption. Control Case 2, apart from the general parameters described in Chapter 4.3., needs the initial set of the dead band radius P_{lim}, and the delay time T_{delay}.

As it is described above, Control Case 2 can be considered as a supplement of Control Case 1, as the sole difference is that the control values of that case are always multiplied by an additional variable K. (see Figure 34.) The calculation flowchart of K is presented on Figure 35. K has a different behaviour in case of the increasing curve (Z=1), and the decreasing curve (Z=-1). The change from one curve to another contains a delay. The corresponding codes for this control can be further studied under Appendix G and H.

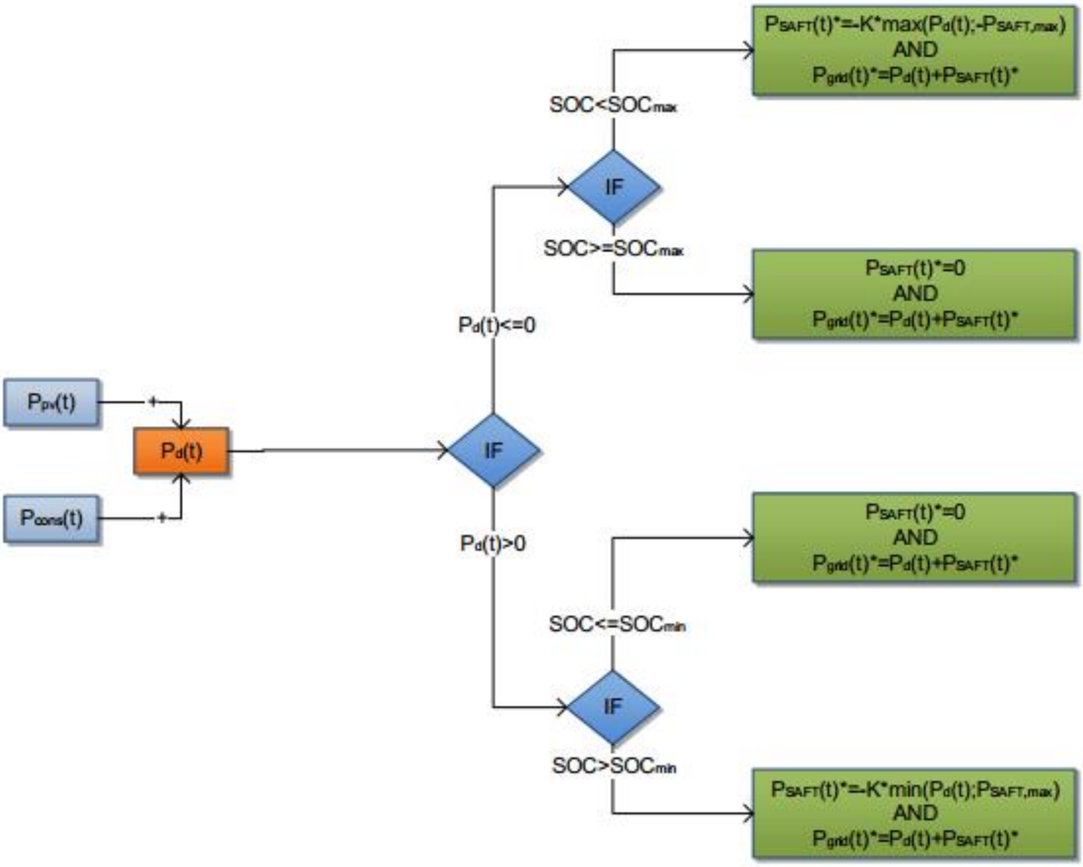


Figure 34: Control flowchart of Control Case 2, part 1.

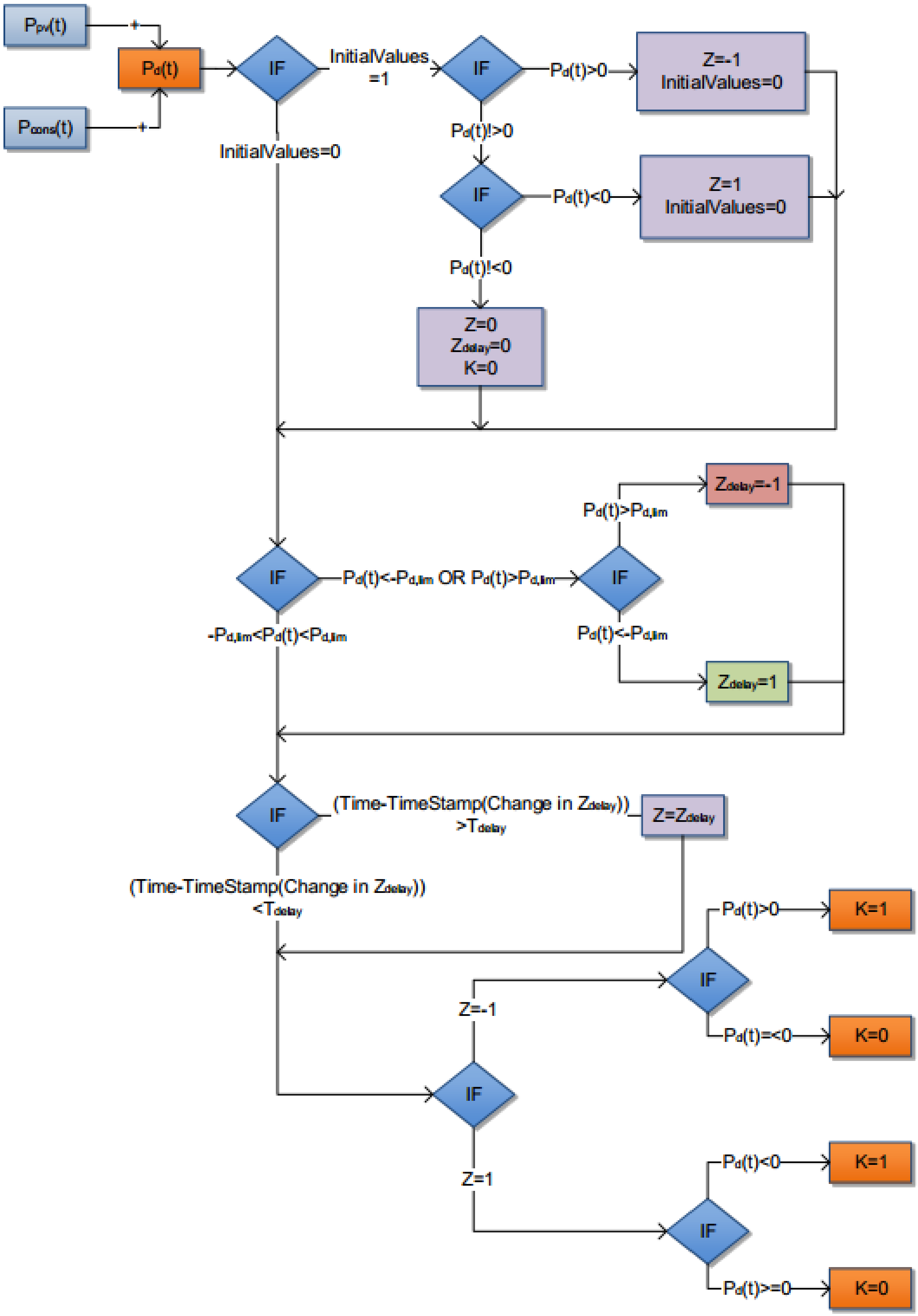


Figure 35: Control flowchart of Control Case 2, part 2.

4.4. Simulation of the showcase in Matlab R2014a

As a preparatory work to the laboratory SCADA test, the above-mentioned showcase with the different control techniques have been simulated in Matlab R2014a environment.

Following the flowchart of the control techniques seen on Figure 32, 34, and Figure 35., the development has solely happened via codes and calculations on the power profile data. Appendix I contains the developed simulation code. With the help of the simulation it is possible to estimate the results of the emulation, and then to see the difference between theoretical and emulated results. The initial SOC has been estimated to be 70%, and then is adjusted when comparing to the emulations. The simplified calculation of the SOC in each second is calculated with the following equation:

$$SOC(t) = SOC(t-1) + \frac{E_{battery}(t)}{E_{battery,max}},$$

where $E_{battery,\,max}$ is the maximum storable energy of the battery (see Appendix B), and $E_{battery}(t)$ is the controlled battery power of each second.

4.4.1. Simulation results of Control Case 1

The results of the simulation of Control Case 1 can be seen on Figure 36. When the cloud arrives, the battery immediately switches from charging to discharging mode. According to the simulation results, the grid is not used at all, as the battery is assumed to reach neither the minimum, nor the maximum SOC, and the needed power is below $P_{saftmax}$ at any moment. SOC has a slight decrease from around 75% to 74% during the period of the cloud. Chapter 4.5.2 shows the results of the real emulation of this type of control.

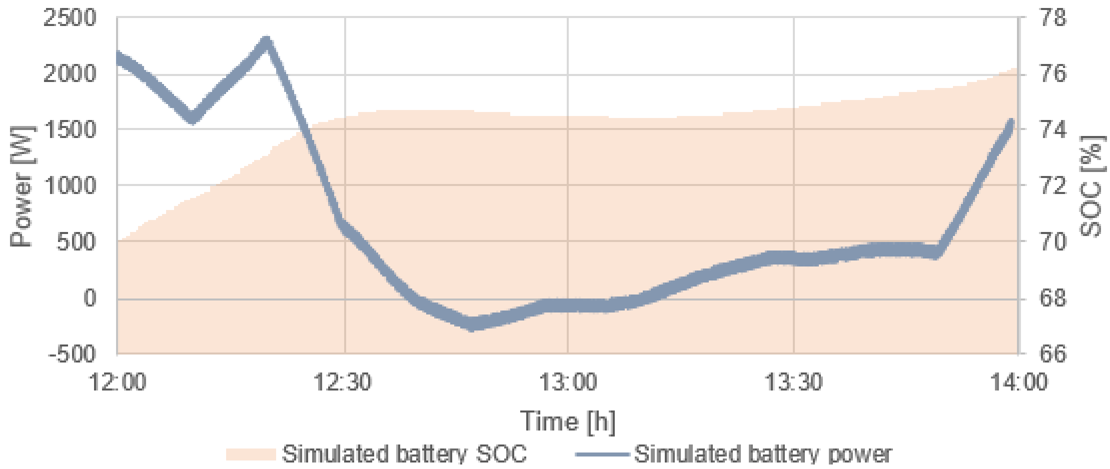


Figure 36: Expected results of Control Case 1

4.4.2. Simulation results of Control Case 2

Control Case 2, additionally to Case 1, has two important parameters: P_{lim} and T_{delay}. Throughout four different simulations, the goal has been to examine the theoretical effect of these parameters. There have been several scenarios run with the parameters introduced in Table 8.

Table 8: Parameters considered during the tuning

Scenario IDs:	P_{lim} [W]	T_{delay} [s]
1	100	60
2	100	120
3	251	120
4	300	120

As the emulation period does not contain consecutive, sudden changes in P_d, T_{delay} results to be not as important parameter for the control. This has been tested by doubling T_{delay} between scenario 1 and scenario 2. It would contain these kinds of sudden changes in case of another gotten power profile, which had more sudden clouds. Nevertheless, T_{delay} should be changed to more than two hours in order to be able to keep 100 W and to mitigate the effect of the cloud. As a result, in Scenario 3 T_{delay} was kept at 120 sec, and the dead band radius P_{lim} was increased to the daily average power demand, 251 W. Figure 37 shows the result of the different scenarios in the most critical time period, when the photovoltaic generation drops to 0 between 12:39 and 13:22. Out of this range, all scenarios the same control behaviour. Changing the dead band radius to the daily average power delays the time more significantly when the battery switches its mode. Nevertheless, the switch has happened at 12:48, when the demanded power dropped below 251 W for over two minutes.

In Scenario 4 an additional surplus was kept, and a dead band radius of 300 W was used in simulation scenario. This scenario has theoretically completely stopped the battery switching to discharging mode. During the critical period the grid is used for covering the needs. This value of P_{lim} is optimal in terms of night use as well. In the evening the base consumption reaches 300 W several times, which turns on the battery usage and facilitates autoconsumption. Figure 37 shows the results of the simulation 4, which has been finally chose to be used during the emulation.

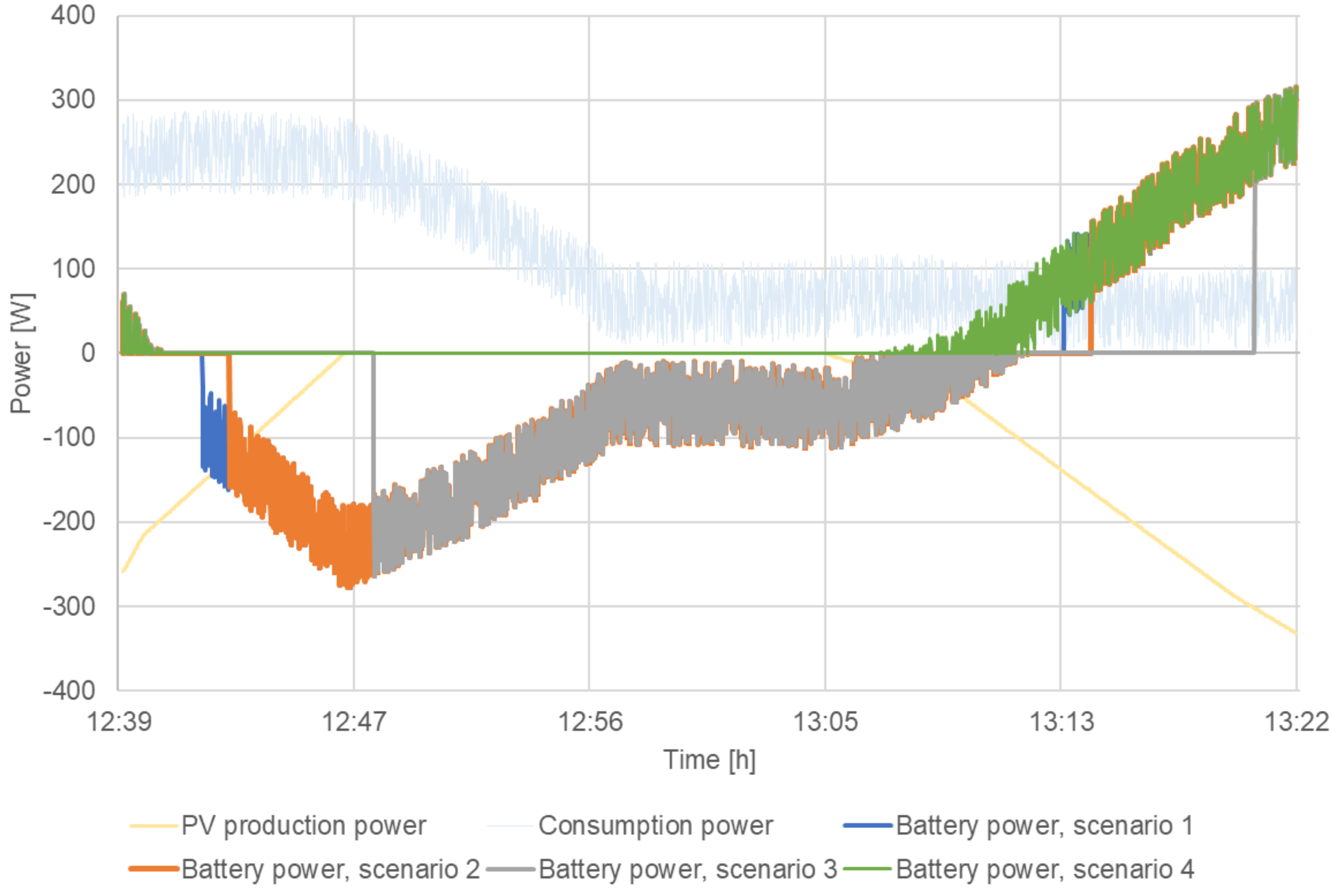


Figure 37: The simulation results for scenario 1 and 2.

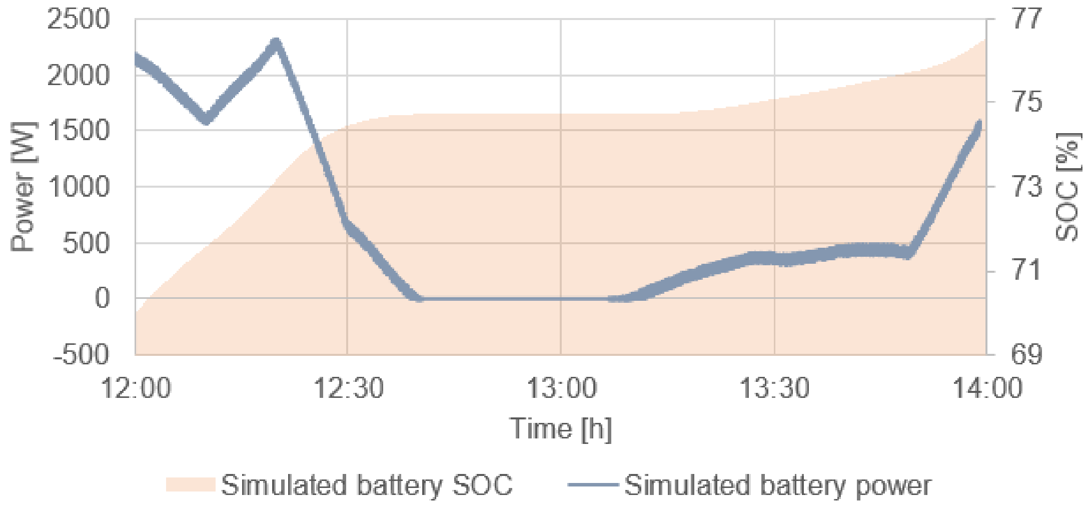


Figure 38: The expected results of Control Case 2 (Scenario 4)

4.5. Emulation tests of the showcase

The emulation showcase of the microgrid SCADA system has been executed on the 5th June 2018 and on 11th June 2018, for Control Case 1 and Control Case 2, respectfully. The following items have been used for the development:

- Emulation cabinet 4, with the LC MYD1 was used for emulating photovoltaic generation.
- Emulation cabinet 6, with the LC RB04 was used for modelling residential consumption.
- The Saft lithium battery with the LC RB16, which was the sole real element of the configuration.

The emulation has the following boundaries:

- During the emulation the power curves of the photovoltaic generation and of the residential consumption are given by pre-defined power profiles loaded to the corresponding LC, giving the values for each second.
- The LC of the battery receives the power setpoints from the SCADA based on the internal control methods explained in Chapter 4.3.
- The power is balanced by the grid during the showcase and is solely calculated throughout the emulation.

4.5.1. Emulation with Control Case 1

4.5.1.1. Event logging

Time: 13th June 2018 – 16:44 – 17:44

Control Case: 1

Events:

- The emulation has started at 16:44. All cabinets started switching normally. The data save was continuous.
- At 16:39 (12:47 of the emulated day) the LC of the battery (RB16) switched to manual

mode. It has been noticed and switched back to auto mode at 16:49 (12.57 of the emulated day)

- At 17:39 (13:47 of the emulated day) the LC of the battery (RB16) switched to manual mode. It has been noticed and switched back to auto mode at 17:43 (13:51 of the emulated day)
- At 17:45 (13:53 of the emulated day) the LC of the battery (RB16) switched to manual mode. It has been noticed and switched back to auto mode at 17:47 (13:54 of the emulated day)
- At 17:44 the demo has stopped all cabinets.

<u>Decision:</u>

The LC of the battery stopped switching three times during the emulation. Nevertheless, as the main power profiles did not get disturbed, and the emulation cabinets did not produce oscillating behaviour (see Appendix K), the test has been decided to get evaluated.

4.5.1.2. Results of the emulation

Figure 39 shows the resulting power curves of the emulated PV generation, residential consumption and of the battery. Moreover, the power setpoints of all cabinets can also be seen in the graph. On the right vertical axis, the current SOC of the battery can be observed as well.

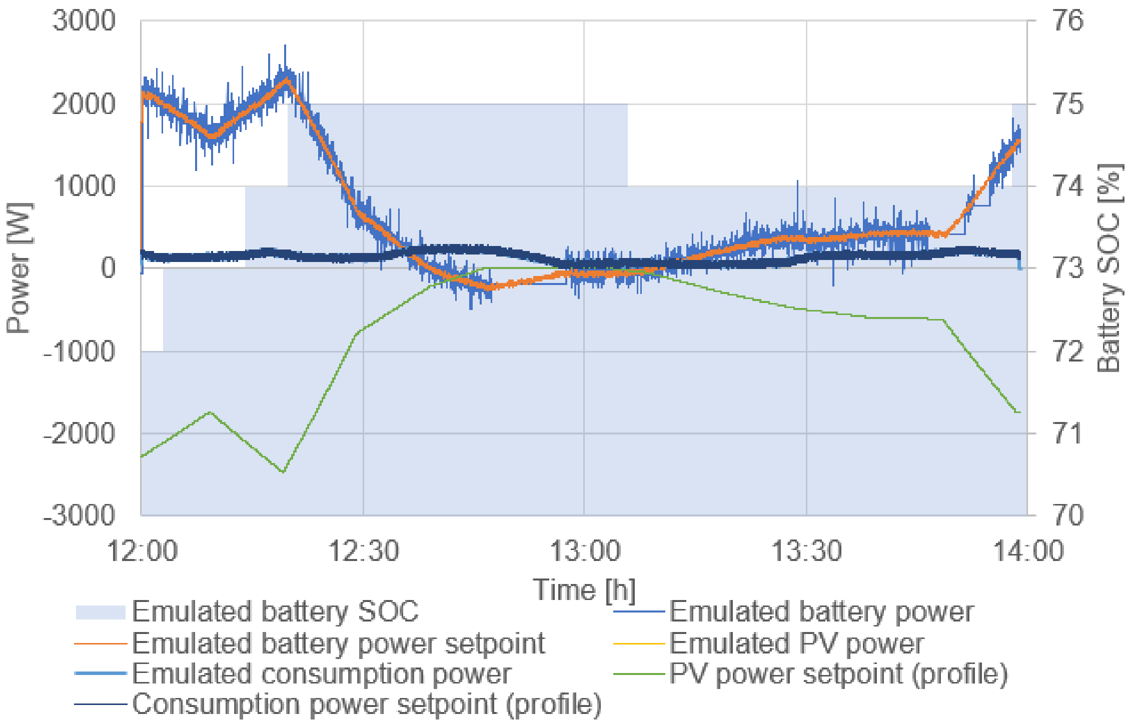


Figure 39: Resulting battery power and SOC curves of the emulation on the 13th June, 2018

The setpoints given to the battery follows well the need at every second. Nevertheless, the exact power given by the battery oscillates around the setpoint. The switch from manual to auto mode can clearly be observed at around 12:47, 13:47 and 13:53 by the flat parts of the measured power. This indicates that although the power setpoints are being calculated in the SCADA control, the cabinet does not switch to the upcoming second.

The SOC of the battery is being calculated in integer form in the cabinet. Being in integer form causes clearly defined rectangles in Figure 38. Nevertheless, as it was expected based on Figure 35, the SOC drops by 1% down to 74% during the period of the cloud.

Control Case 1 solely sets the grid power other than 0 when the SOC or P_{demand} are out of the normal conditions. After conducting the simulation, and then the emulation, it can be seen that it did not happen throughout the two hours of testing. Hence, the error of the control system comes from the following equation:

$$P_{error}(t) = P_{grid}(t) = P_{PV,meas}(t) + P_{cons,meas}(t) + P_{saft,meas}(t)$$

After calculating the error for each second, the data has been organized in histogram format of 50 W range. Afterwards, the histogram has been normalized to probability density. The resulting graph can be seen in Figure 40. The diagram shows that more than half of the times the error is lower than 50 W. If the tolerance is 100 W, then around 80% of the measurement points are considered normal. Nevertheless, the main error source of this emulation has been the three cases of manual mode switch. These error range is the reason behind [-150;100] period has a higher probability density than [50;100) As it can be seen on the histogram, the probability density of an error exponentially decreases with the magnitude.

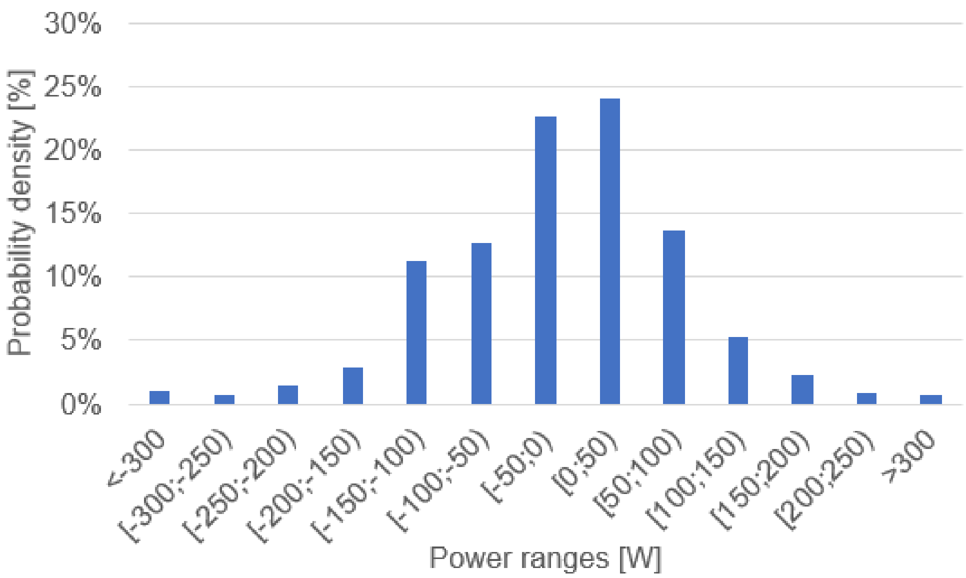


Figure 40: Probability density of error ranges of the emulation on 13th June, 2018

4.5.2. Emulation with Control Case 2

4.5.2.1. Event logging

Time: 11th June 2018 – 16:36 – 18:36

Control Case: 2

Events:

- The emulation has started at 16:36. Emulation cabinet no. 4. (with LC RB04, PV emulation) did not switch to auto mode in the first minute. After several tries, in 16:37 the cabinet started the switching in auto mode.
- At 17:00 (12:24 of the emulated day) emulation cabinet no. 6 (with LC MYD1, consumption emulation) switched to manual mode. The event was noticed immediately and at 17:02 (12:26 of the emulated day) the cabinet continued switching.
- At 18:36 the demo script of the SCADA has stopped all cabinets.

Decision:

Apart from one switch to manual mode, the cabinets functioned well. The test has been decided to get evaluated.

4.5.2.2. Results of the test

Figure 41 shows the resulting power curves of the emulated PV generation, residential consumption and of the battery. Moreover, the original curves of PV generation and of consumption can be seen in the graph. On the right vertical axis, the current SOC of the battery can be observed as well. It can be seen that control method in the SCADA successfully avoided the switch to battery discharging setpoints. During the presence of the cloud, there are no negative power setpoints sent. Nevertheless, as the exact injected powers vary around the power setpoint, there are numerous seconds in which the battery is getting discharged during the two hours of the test. This, apart from the proportionally higher losses of the power electronics in case of lower injected battery power, is the reason why the battery SOC remained 71% during most of the test. It means a final difference of 4% compared to the simulation of Control Case 2, in which the SOC has reached nearly 75%.

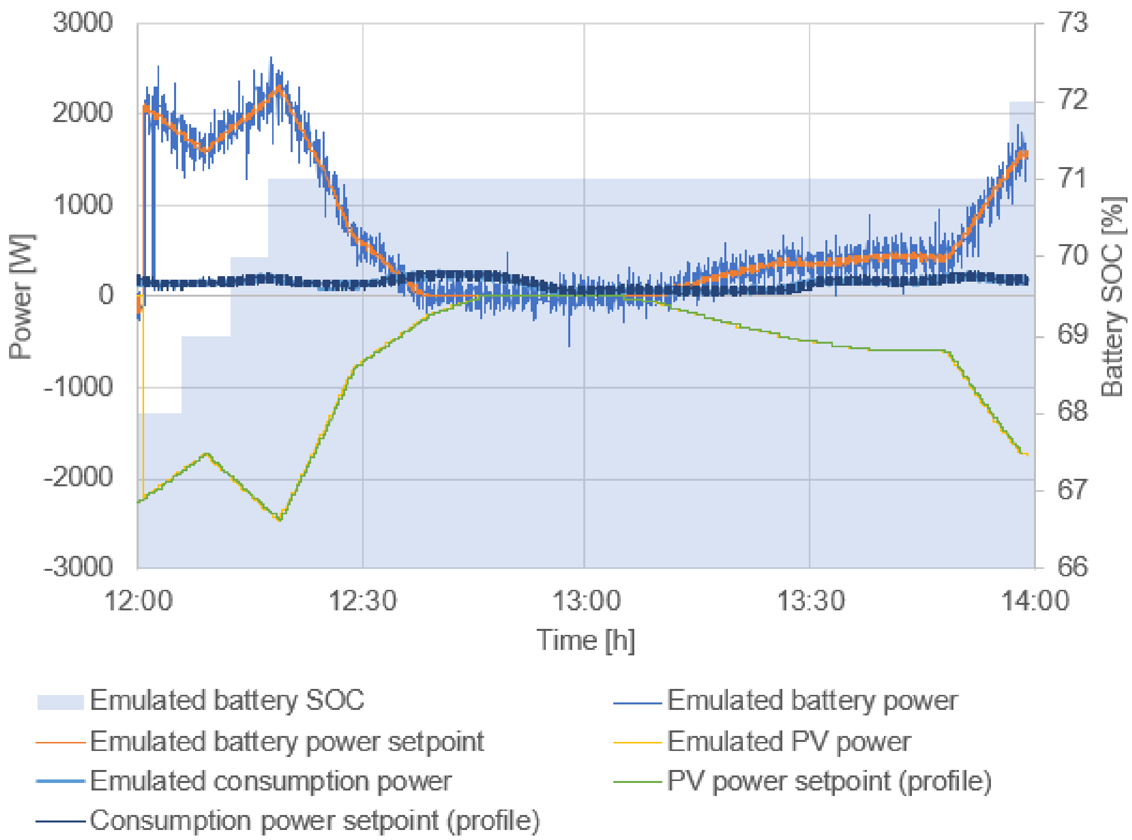


Figure 41: Resulting battery power and SOC curves of the emulation on the 11th June

The comparison of the two curves can be seen in Figure 42. The upper-mentioned assumptions for the explanation for the SOC difference is supported by the fact that the emulated and the simulated power setpoint curves do not differ significantly. Regarding the exact difference, Figure 43 exposes more information. It shows the comparison of the simulated and of the emulated battery powers in the critical period of when the cloud arrives and leaves. The power setpoint curves of the battery slightly differ, which could come from the variation of the two other emulations around their setpoints. Nevertheless, the highest variation is produced by the real emulated battery power.

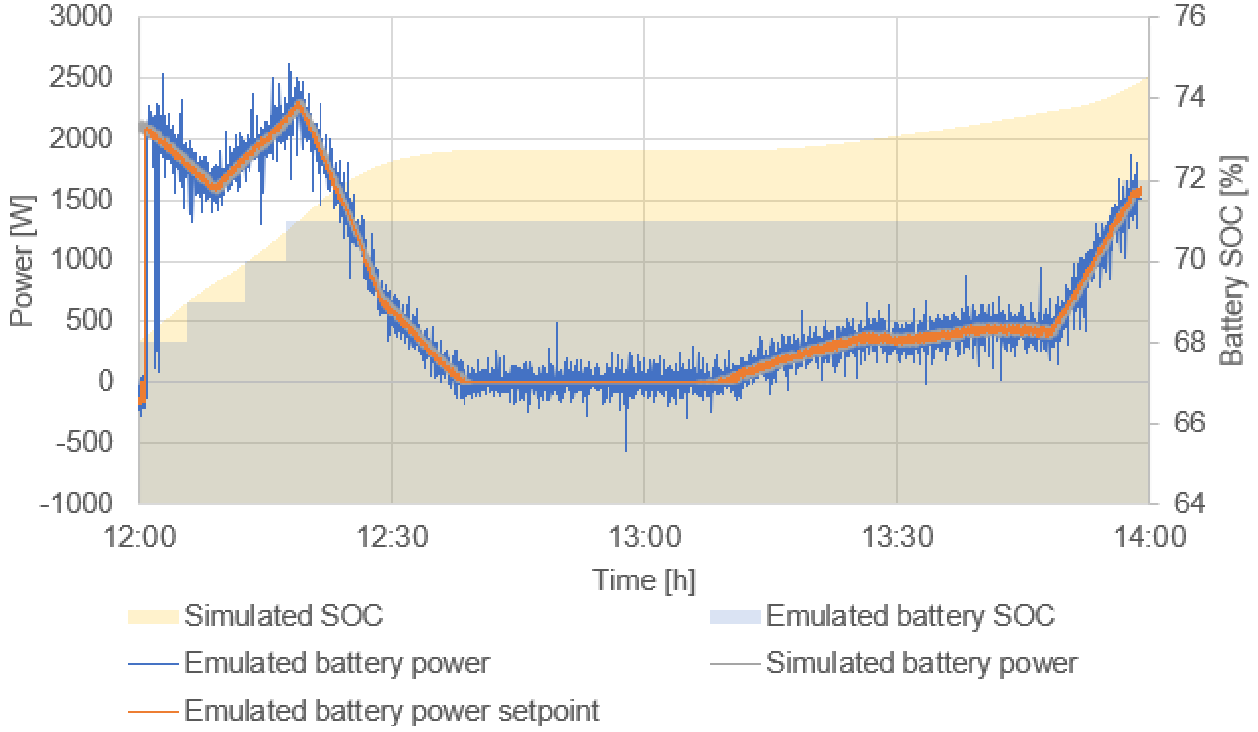


Figure 42: Simulated and emulated battery power and SOC during the test

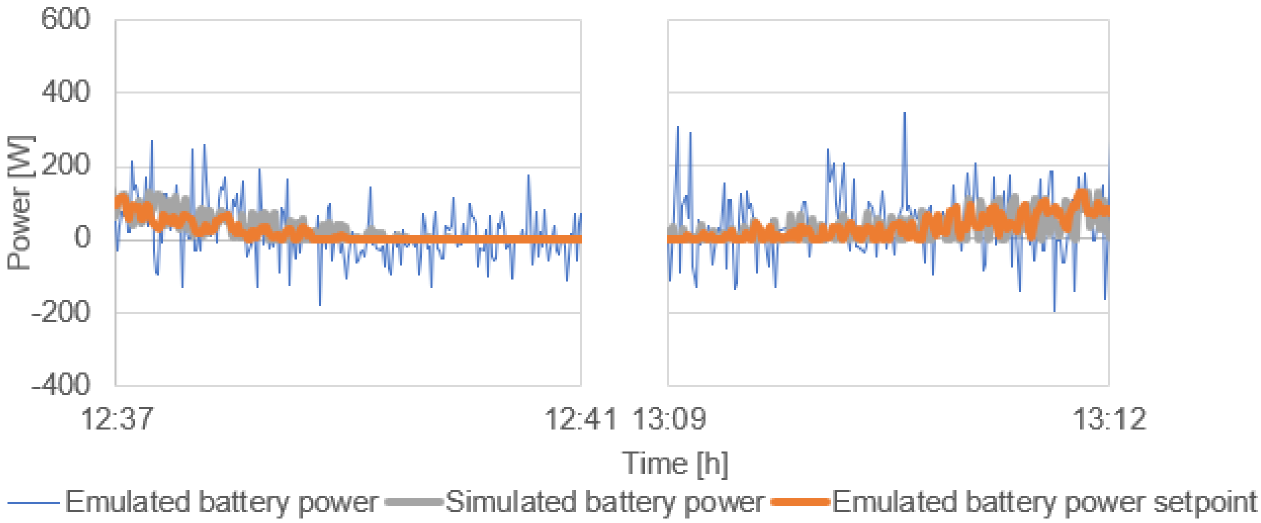


Figure 43: Behaviour of the emulation and of the simulation when the cloud arrives and leaves

4.5.3. Cabinet Accuracy

Emulation cabinets are suitable for conducting experiments, nevertheless, their capability to follow the upcoming setpoint has some inaccuracy. The following chapter analyses the capability of the three cabinets (no 3, 4, and 6) to follow their setpoints. In a chosen zoomed period, from 12:10 to 12:15 pm the curves of cabinet setpoints and the measured powers are presented for all three cabinets.

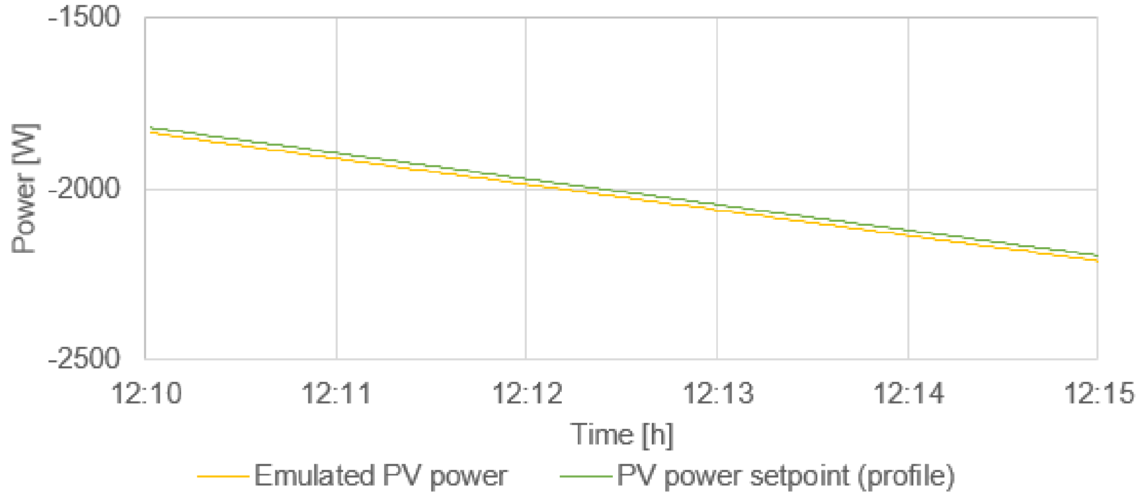


Figure 44: Setpoint follow of emulation cabinet no. 4. (with LC RB04) – photovoltaic emulation

Figure 44 shows the curves of emulation cabinet 4 (with LC RB04), which was used for photovoltaic emulation during the showcase. In the zoomed period, the setpoint is strictly monotonically decreasing from around -1700 W to -2300 W. As it can be seen, the linear trendline of the actual generation power is parallel to the setpoints, nevertheless, it is below it by around 20 W constantly. This probably comes from the fact that RB04 did not switch immediately to auto mode, meaning a bit of delay in the power profile.

Figure 45 shows the curves of emulation cabinet no 6. (with LC MYD1), which was used for consumption emulation. The consumption profile highly varies at every second around its trend. There are numerous points of this curve in which a delay can be seen from the setpoint to the actual measurement points. For example, the minimum points of 100 W at around 12:10:20 reflect in the measurement after around 8 seconds. Another point where this delay can be observed is at 12:14:45, when the power setpoints reach a maximum, and the decrease by 50 W immediately. The measured power follows this profile around 7-8 seconds later, when the next power setpoints are already in the valley. Nevertheless, it can be generally seen that the exact power setpoint is hardly achieved because of the high variation.

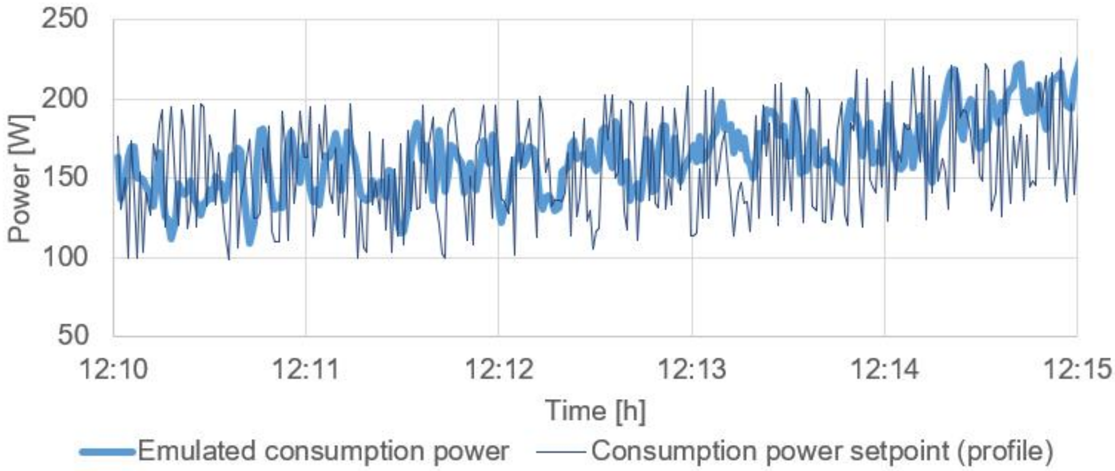


Figure 45: Setpoint follow of the battery, cabinet no 6. (with LC MYD1) – consumption emulation

Based on the measured powers of the consumption and of the production, the SCADA calculates the power setpoints to be sent to RB16, the LC of the Saft battery. The photovoltaic production shows a stabile decrease during these 5 minutes, nevertheless, the consumption emulation highly varies. This causes the little disturbances in the given power setpoints. Nevertheless, it can be seen that the actual measured power highly varies around the power setpoint, often with more than 500 W for a second. This can be seen in Figure 46. These disturbances remained the same after triggering both the SCADA control and the Modbus TCP/IP write functions. The phenomenon indicates that the battery has the highest variance around a given setpoint out of the three elements.

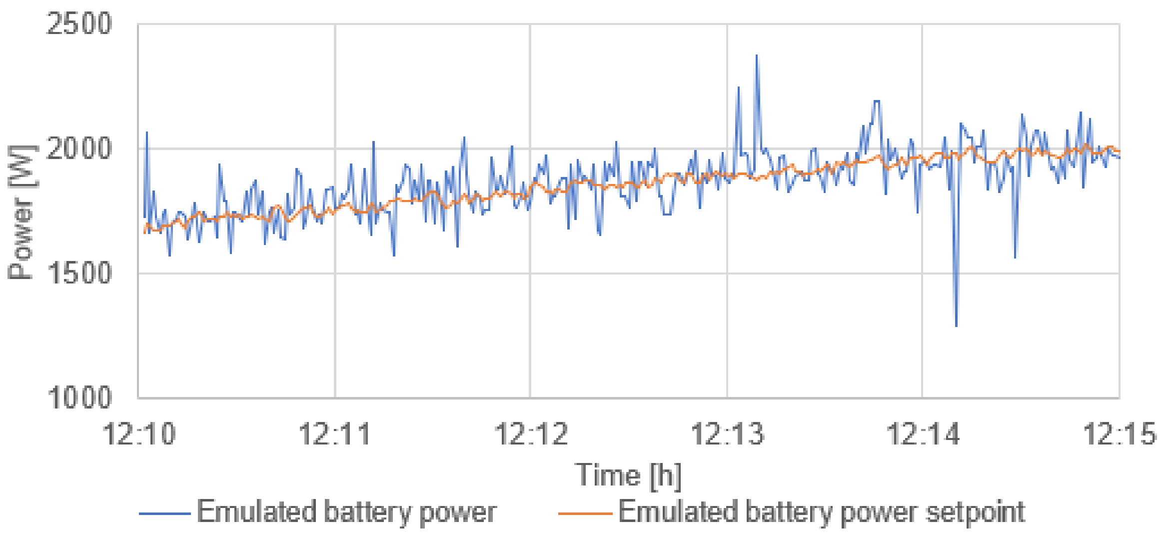


Figure 46: Setpoint follow of the battery, cabinet no 3. (with LC RB16)

Figure 47 and 48 show the statistical difference between the power setpoints and real powers in a histogram form. Figure 45 presents the absolute errors in 50 W ranges. Analyizing Cabinet 4, it produces 100% of its errors below 50 W. Nonetheless, this cabinet was responsible for the PV emulation, hence, the lowest power differentials. 100% of the consumption cabinet

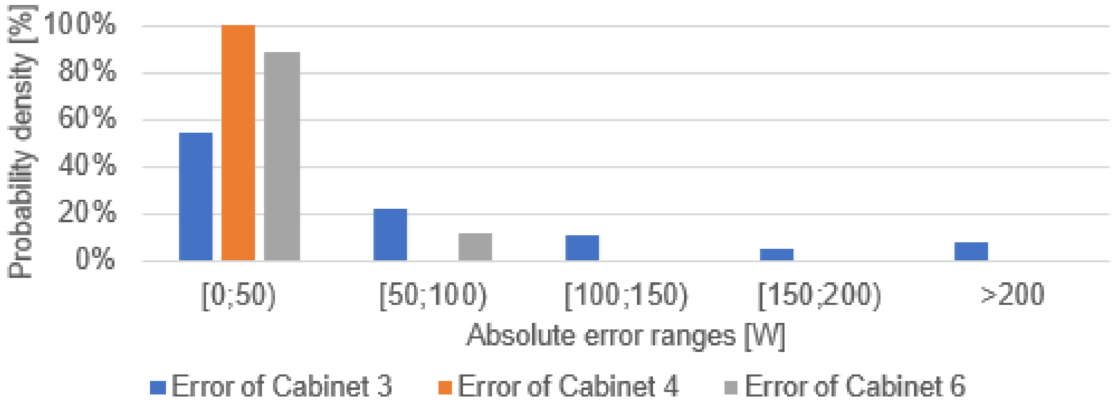


Figure 47: Absolute errors of the cabinets

errors, Cabinet 6, is below 100 W. Nevertheless, having proportionally lower power values, it produces the relatively highest errors. This phenomenon can be seen in Figure 46 which presents the relative errors of the cabinets. As it can be seen, in absolute terms the cabinet of the battery produces the highest errors, having almost 10% of its errors over 200 W. Figure 41 shows the power values around the period when the power is set 0 constantly for more than 15 minutes. Although the setpoint does not change, the power given by the battery varies. It means that the error of the cabinet does not depend on the constant change in the power setpoint. Further experiments are needed for the validation of these cabinet errors. This could be reached by different power profiles, changing the roles of the cabinet, or by a different power control of the battery, e.g. a control with constant power setpoint values.

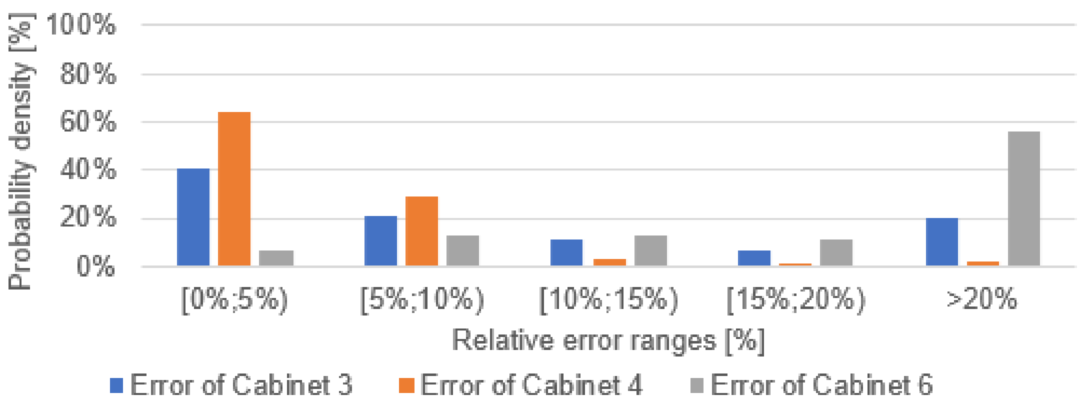


Figure 48: Relative error ranges of the cabinets

5) Innovative Project Idea for prosumer billing

The aim of all microgrid laboratory emulations is make the transition to a decentralized power system technically feasible. Nevertheless, a decentralized power system would not only imply technical, but business challenges as well. The following chapter proposes an innovative solution for one of these challenges, the integrated billing of prosumers and EVs. The proposed prototype project is called *Smartricity*, being developed as a scalable intrapreneurial idea for an energy utility in Barcelona.

5.1. Problem description

The current consumer electricity market is rather static, and very linear. Intraday pricing is emerging, but not widespread, and mostly limited to rather elementary off-peak/on-peak schemes. Communication between consumers and their suppliers is often limited to the yearly electricity bill. However, microgrids, EVs, IoT solutions and intelligent homes are getting more and more widespread, and are on the brink of enabling two-way, real-time communication between energy utilities and their customers.

5.2. Innovative Idea Description

Smartricity would fit the most to an intrapreneurial business model innovation of energy utilities, enabling an additional business model for their portfolio. The value proposition of the idea lies in three central elements for a disruptive application of blockchain to the emerging microgrid and EV revolution:

1. The introduction of a digital *Smartricity App*, integrated electricity hub and billing system, which will be able to establish two-way communication between the utility and the electricity-consuming devices of its customers.
2. Instalment of utility-owned EV-chargers. EV owners who would like to charge their vehicle use these stations, identifying themselves using a customizable *Smartricity App*, and start the charging process – while their payment is completed through the *Smartricity App*, using the utility-provided electronic wallet based on blockchain.
3. The disrupted consumer-energy utility interaction facilitates a completely new business model – minute-resolved intra-day pricing which consumer devices autonomously react to in real-time. This opens saving opportunities for customers, as they can e.g. instruct their HVAC to run predominantly during low-price times. For the energy utility, this means an entirely new form of market-based load control and adaptive pricing possibilities, both of which can generate entirely new business opportunities.

5.3. Initial Business Model (BM)

As a starting model of the upper-described horizon, the initial business model focus on the EV customers. For testing and optimization purposes of the machine-to-machine contacts, as well as the optimization of controllable residential loads, IREC's SmartLab laboratory facility could be used. The laboratory, with the implemented SCADA, EMS, and the cabinets is suitable for testing control algorithms of residential batteries, PV panels, and electric vehicles. Apart from EV charging, these types of customizable controls would be the core value proposition of the *Smartricity*, run on blockchain.

Figure 49 shows the Business Model Canvas of the project. *Smartricity* business model consists of two main value propositions: it combines the billing process of EV charges and the domestic electricity transfer via a blockchain-based electronic wallet. Giving further opportunities for machine-to-machine interactions and B2C-pathways, energy prosumer two-sided exchanges and EV charges would appear in one single electronic wallet.

The initial customer segment of a *Smartricity project* could be the increasing number of those EV owners who would like to be pioneers in an easy and environmental-supportive EV-charging solution. The rise of domestic electricity production can lead to various customer relations. For example, the electronic wallet could give the opportunity for prosumers to indicate their support for further sustainable development projects. The idea is to create a modern, customizable *Smartricity* App by which the communication between the utilities and the prosumers' electronic wallets could happen directly and seamlessly. Another form of channel is via the EV charger stations. The identification of the customers at the charging stations could easily happen with an ID built in the personalized *Smartricity App*, connected to the ID details of the customer. At the beginning of a charging session, after the customer identification, sets the details of the charge and then the battery of the vehicle can already start the filling process. When customers are at home, the *Smartricity App* would help them manage their EV-charging.

The energy utility already has a developed customer segment. These customers could get notified about the *Smartricity App* via a simple letter. Those early adopters who join, charge their EV and start to control their prosumption via the app, could get special initial discount. The participating EV charger stations, parking lots would receive a sustainable promotion by the utility on the streets, social media etc. They would be placed in optimal parking lots, and hence, they could attach further customers to the service.

The business runs in collaboration with its key partners. Namely, the government, Barcelona municipality, parking lot owners, IREC's SmartLab Team and a blockchain provider IT company. As a sustainable smart solution, the project could be supported by the European Union, as well as by governments, especially under Horizon 2020. Currently, EV charges are paid by these entities, hence, they could be highly interested in setting up a sustainable system for the payments, having advantages for all participating parts. As mentioned earlier, EV charger operators and parking lot owners would gain a sustainable promotion, and extra traffic by partnering with Smartricity project. IREC's SmartLab could be the perfect research partner for testing and optimization of the machine-to-machine contacts. Last, but not least, a specialized blockchain provider IT company would be partnered up for providing the core IT solution, as well as potentially the operation of the blockchain.

As key initial activity and cost, the utility would contract a blockchain-specialized company for the development of the technological background of the application (electronic wallet), as well as for the establishment of EV charger stations or for contacting currently existing networks. Afterwards, operating the system would consist of running the application, carrying out the two-sided electricity exchanges and storing them with blockchain technology.

The project would mainly sustain itself by the consumers who pay for their electricity used for both transport and domestic level. As a smart development, *Smartricity projects* could be entitled for financial support by governmental entities, giving extra sources of revenue. In case of intra-day pricing of the electricity charger stations, consumer choices could help the utility to stabilize their electricity supply and demand. Thanks to the dynamic and fast mechanism supported by blockchain technology, the grid stability would increase, generating an aspect of decreasing overall operating expense for the utility.

The Minimum Viable Product (MVP) of the upper-described business model would be Barcelona. This vibrating city is not only the home of numerous potential EV owner customers, but has various local blockchain companies, and research institutes in the field. This MVP would give the opportunity for testing, optimizing, as well as observing the possibly changing customer behaviour. In case the project gets successful, the energy utility could move the project to other city and would gain a high competitive advantage by its gained expertise and trust by the people.

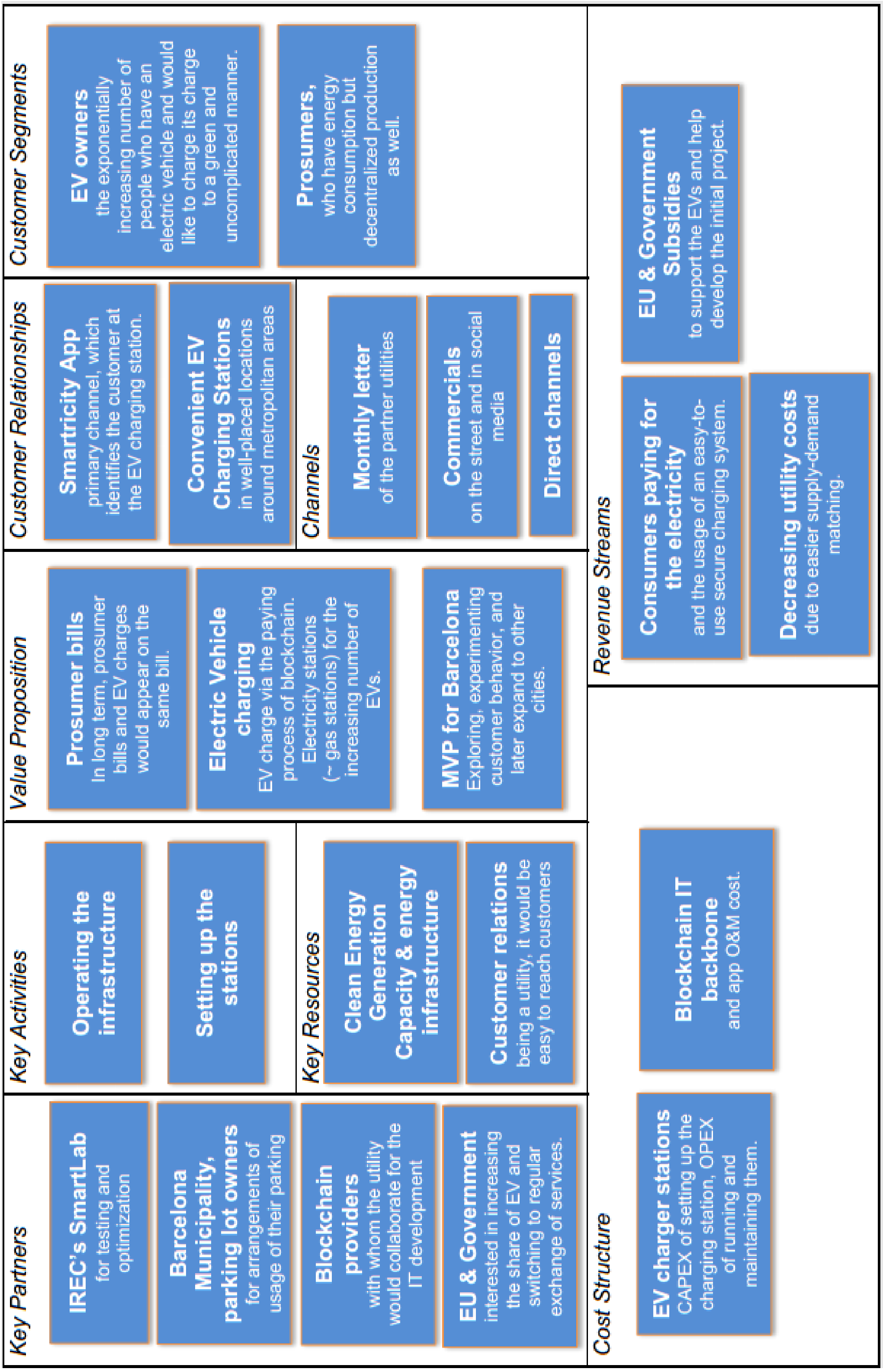


Figure 49: Initial Business Model Proposal for the Innovative Project Idea

6) Conclusions

I have had the opportunity to work 5 months on the SCADA design and development, testing for IREC's SmartLab within the master thesis development of the EIT InnoEnergy's Renewable Energy Master program.

Chapter 1.1. has showed the general importance of SCADA systems for the integration of microgrids and put the evolution of microgrid projects in perspective. Moreover, Chapter 1.2. and 1.3. there have been both qualitative and quantitative research conducted about the state of the art and the global market drivers of microgrids. After the outlook of the regulatory aspects in Chapter 1.3.3., Chapter 2. gives an in-depth review of IREC's SmartLab facilities and its plans. The laboratory has been examined both from electrical and from communicational point of view including the main components, such as the cabinet LCs, and the management roadmaps of the configuration and of the operation modules.

After the initial reviews, Chapter 3. describes the core of the project: the developed SCADA system for the laboratory. All the requirements described in Chapter 3.1. have been reached in the developed 7 screens. Throughout the development of the SCADA system there have been several lessons learnt. Amongst others, the following ones should be highlighted:

- The configuration and the operation modules of the microgrid laboratory should strongly define the design of the SCADA. The modules should be shown as a working environment for the user.
- The project symbols of ITME are useful for making similar objects easily editable at once, as well as showing all the information needed about several similar devices.
- Classes of ITME are suitable for storing information about devices with similar type of configuration, such as of LCs.
- The SCADA should send control signals in the same rhythm as it receives information, otherwise there could be communicational problems and false control signal sent.
- The demo screen is always specific to the studied case and cannot be made in a general form.

The developed SCADA system improved the monitorability and controllability of the microgrid laboratory. By accessing all the LCs from one program, it is no more needed to physically be close to all of them at once. It is possible to read all the necessary data for the operators, as well as to control them in case of any actions needed. This initial development facilitates running emulations on the system with high transparency and controllability. As such, it fits in the picture of the laboratory management plans, described in chapter 2.3.3.

To test the system, a showcase simulation and emulation were run, described in Chapter 4. The simulation has been executed in Matlab R2014a environment, and the emulation on IREC's microgrid with facilitated by the developed SCADA. There have been numerous observations made when putting a simulation in the reality of an emulation with real power flows. As one of the main conclusions of this part of the work, the internal control of the Saft battery does not allow it to follow as accurately the sent setpoints, as the other cabinets. Nevertheless, none of the cabinets are capable of following a setpoint as much as it happens throughout a simulation.

The two developed algorithms have resulted sufficient for the needs of this project. Control Case 1 shows a basic real-time control of the battery, and Control Case 2 has solved supplemented it with a hysteresis curve. Nevertheless, for real optimization of autoconsumption, more sophisticated control systems are needed, for which directions are pointed out in Chapter 6.2.2. When the laboratory fully operates with the EMS and the SCADA installed, it could be used for several types of projects in collaborations with several stakeholders of the energy industry. An interesting business model has been proposed in Chapter 5 for an energy utility which could collaborate with IREC's laboratory facilities. The business model shows the complexity of a whole project in which these facilities would be inevitable for optimization and tests.

6.1. Challenges and solutions

There have been numerous upcoming challenges during the development of the project. Amongst others, the following points made this master thesis a real challenge:

- Several technical challenges related to profile configurations, such as the dynamic change of profile names in the configuration screen. This issue has been tackled by the introduction of two, independent profile configuration scripts.
- The LCs are currently under development and new configurations are being implemented. It has caused a certain amount of uncertainty during the project. It has been solved by designing the SCADA (e.g. state machine) for the new LCs but conducting the showcase for the old LCs.
- During the showcase, the LCs lost contact several times with the SCADA. It has been solved by increasing the retries of the driver configurations and introducing a trigger-based writing scheme instead of a constant MB read-write.
- During the emulations, there has been momentary communicational errors observed, causing instable behaviour. These were not saved because of the different triggering of the database and of the Modbus TCP/IP communications. It has been solved by triggering the SCADA control as well. Later it shall be solved by working with average values.
- During the showcase experiment, Control Case 2, the control parameters highly influence the results, nevertheless, there are limited times to run the emulation. It has been solved by building up the models in Matlab as well and by simulation estimate

the suitable parameters.

- Being a practicable, real-world project for research purposes, it has been a challenging task to relate the laboratory project to business models. It has been solved by introducing a possible intrapreneurial project idea for an energy utility, in which IREC's SmartLab would participate as a partner.

6.2. Future works

Future research in the laboratory lanes range from general improvements of the SCADA work to specific project demo developments. Regarding specific and expected showcase developments the following directions of future work have been identified:

- The conductance of the showcase experiment with more instable weather and consumption conditions could be recommended for future experiments.
- Inclusion of charge-discharge curves, and an ageing factor of the battery in the control loop of the Saft battery.
- In industry control systems usually do not work with real-time values because of drawbacks listed in Chapter 4.4.1. A hysteresis and pre-defined control value-based control proposal with a flowchart shown in Appendix J has been designed for further testing. The simulation and emulation of this control could be an initial step for further improving the project.
- Lastly, with the introduction of the new LC configurations, EMS management, it is suggested to conduct the autoconsumption showcase with price-dependency, and varying auto or manual mode of the laboratory.

As regards to general recommendations for the SCADA improvement, the following suggestions have been made:

- As soon as the new LC configurations are ready, amongst others, the following points are important to consider:
    - The prepared SCADA state machine algorithm should be precisely matched and tested with the LCs' ones.
    - FTP connection shall be established between the elements and the SCADA so that adding a new profile curve could be executed in a fast and easy way.
    - As the configuration and the operational ports would merge, there should be one ultimate driver sheet instead of the currently used two ports-system.
- The specific control and supervisory board of the grid emulator should be developed and tested.
- Further parameters could be considered for saving in MySQL and being configured as historical data.
- All objects used for profile visualization should be revised in terms of load speed.
- The creation of other Demo pages corresponding to further upcoming IREC projects is also expected.

ACKNOWLEDGEMENT

I would first like to like to say special thanks to my supervisor, Alba Colet for the weekly help and support. She always steered me in the right direction whenever she thought I needed it. Her professionalism, supportive and educative attitude have helped me immensely towards the successful realization of this project.

I would like to thank all the Power Electronics and Electric Power Grids Group of IREC, whose valuable comments, inspiring environment have been truly important for the successful results of these 5 months. I hope to be able to work with you and see all of you again one day. I would like to say special thanks to Guillermo Domínguez, whose helping attitude and expertise at InTouch Machine Edition software have provided indispensable help during the development of the project. Moreover, I would like to highlight the continuous help and revision of Cristina Corchero, Pol Paradell and to José Luís Dominguez. Thank you!

I would also like to acknowledge Prof. Oriol Gomis Belmunt from the Department of Electrical Engineering (CITCEA) of the Universitat Politècnica de Catalunya, whose comments on the thesis and knowledge in SCADA systems for microgrids have been highly valuable during these months.

I would also like to thank EIT InnoEnergy Master School, whose prestigious program and generous scholarship have made it possible for me have this double degree study track on Renewable Energy starting at École Polytechnique, University of Paris-Saclay, finishing at Universitat Politècnica de Catalunya (UPC).

Finally, I must express my very profound gratitude to my parents, my family, and my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of conducting this master thesis. Mostly, I would like to say thanks to my parents, without whom I would not even be here. Thank you!

Benedek Pásztor

Bibliography

[1] IEA, Energy Access Outlook 2017, Paris: International Energy Agency, 2017.

[2] Z. X., Z. G., G. J. M. and H. F., “SCADA system for islanded DC microgrids,” in *IECON 2017*, Beijing, 2017.

[3] IRENA, “REthinking Energy 2017,” International Renewable Energy Agency, Abu Dhabi, 2017.

[4] DOE, “How microgrids work?,” 2014. [Online]. Available: https://www.energy.gov/articles/how-microgrids-work. [Accessed 4 May 2018].

[5] I. Alsaidan, A. Alanazi, W. Gao, H. Wu and A. Khodaei, “State-Of-The-Art in Microgrid-Integrated Distributed Energy Storage Sizing,” *Energies,* p. 10, 2017.

[6] Microgrid Investment Accelerator, “Microgrid Market Analysis & Investment Opportunities - India, Indonesia, and Tanzania,” Microgrid Investment Accelerator, 2016.

[7] M. Agrawal and A. Mittal, “Micro grid technological activities across the globe: a review,” *Ijrras,* 2011.

[8] S. Parhizi, H. Lotfi, A. Khodaei and S. Bahramirad, “State of the Art in Research on Microgrids: A Review,” *IEEE Access,* 2015.

[9] M. Roman-Barri, I. Cairo-Molins, S. A. and A. Sudria-Andreu, “Experience on the Implementation of a Microgrid Projet in Barcelona,” in *ISGT Europe*, Gothenberg, Sweden, 2010.

[10] R. Enrich, P. Skovron, M. Tolós and M. Torrent-Moreno, “Microgrid management based on economic and technical criteria,” Barcelona, 2013.

[11] Consortium for Electric Reliability Technology SOlutions, “CERTS Microgrid Concept,” [Online]. Available: https://certs.lbl.gov/initiatives/certs-microgrid-concept. [Accessed 20 05 2018].

[12] K. Thu Zar Win and H. Myo Tun, “Design and Implementation of SCADA System Based Power Distribution for Primary Substation,” *IJECSE,* 2014.

[13] Lee, E-K., Shi, W., Gadh, R., Kim, W., “Design and Implementation of a Microgrid Energy Management System,” *Sustainability,* 2016.

[14] T. S. Ustun, C. Ozansoy and A. Zayegh, “Recent developments in microgrids and example cases around the world - a review,” *Renewable and Sustainable Energy Reviews,* 2011.

[15] L. Cardwell and A. Shebanow, “The Efficacy and Challenges of SCADA and Smart Grid Integration,” *Journal of Cyber Security and Information Systems,* 2016.

[16] N.-G. Park, M. Gratzel, T. Miyasaka, K. Zhu and K. Emery, “Towards stable and commercially available perovskite solar cells,” *Nature Energy,* 2016.

[17] D. Kumar, S. Kumar Rajouria, S. B. Kuhar and D. K. Kanchan, “Progress and prospects of sodium-sulfur batteries: A review,” *Solid State Ionics,* 2017.

[18] T. L. Nguyen, Q.-T. Tran, R. Caire, C. Gavriluta and V. H. Nguyen, “Agent Based Distributed Control of Islanded Microgrid - Real-Time Cyber-Physical Implementation,” Torino, 2017.

[19] A. Khaledian and M. A. Golkar, “Analysis of droop control method in an autonomous microgrid,” *Journal of Applied Research and Technology,* 2017.

[20] C. Li, C. Cao, Y. Cao and B. Fang, “A review of islanding detection methods for microgrid,” *Renewable and Sustainable Energy Reviews,* 2014.

[21] C. K. and J. H. C. Huang, “Fault detection, classification and location for transmission lines and distribution systems: a review on the methods,” *IET,* vol. 1, no. 1, pp. 25-33, 2016.

[22] P. Aristidou, A. Dimeas and N. Hatziargyriou, “Microgrid Modelling and Analysis Using Game Theory Methods,” *Energy-Efficient Computing and Networking,* pp. 12-19, 2010.

[23] D. T., W. D., S. Q. and M. L., “Distributed and Decentralized Control Architectures for Converter-Interfaced Microgrids,” *Chinese Journal of Electrical Engineering,* vol. 3, no. 2, 2017.

[24] A. Aram, “Microgrid Market in the USA,” Global Innovation Report, Tarrytown, 2017.

[25] IMARC Group, “Microgrid Market: GLobal Industry Trends, Share, Size, Growth, Opportunity and Forecast 2018-2023,” IMARC Group, Noida, 2017.

[26] T. Samad, E. Koch and P. Stluka, “Automated Demand Response for Smart Buildings and Microgrids: The State of the Practice and Research Challenges,” *Proceedings of the IEEE,* vol. 104, no. 4, pp. 726-744, 2016.

[27] Visiongain, “The microgrid market forecast 2017-2027,” Visiongain Ltd., London, 2017.

[28] ABB, “Microgrids Overview Europe and Africa,” Newcastle, 2017.

[29] Navigant Research, “Microgrid Deployment Tracker 2Q17,” London, 2017.

[30] ENEA Consulting, “Microgrids for commercial and industrial companies - delivering increased power reliabililty, lower energy costs and lower emissions,” World Business Council for Sustainable Development, Geneva, 2017.

[31] D. Schnitzer, D. S. Lounsbury, J. P. Carvallo, R. Deshmukh, J. Apt and D. M. Kammen, “Microgrids for Rural Electrification - A critical review of best practices based on seven case studies,” United Nations Foundations, Paris, 2014.

[32] DESI Power, “DESI Power website,” 2018. [Online]. Available: http://www.desipower.com. [Accessed 30 05 2018].

[33] Enea Consulting, “Urban Microgrids - Overview, challenges and opportunities,” Paris, 2017.

[34] Maitra, A.; York, B.; Kamath, H.; Key, T.; Singhvi, V., “Microgrids: A Primer,” Electric Power Research Institute, 2013.

[35] European Council, “Directive 2009/72 EC of the European Parliament and of the Council,” *Official Journal of the European Union,* 2009.

[36] A. Colet-Subirachs, A. Ruiz-Álvarez, O. Gomis-Bellmunt, F. Alvarez-Cuevas-Figuerola and A. Sudriá-Andreu, “Centralized and Distributed Active and Reactive Power Control of a Utility Connected Microgrid Using IEC 61850,” *IEEE Systems Journal,* pp. vol. 6, no. 1, 2012.

[37] ESA, "Energy Storage Technologies," 2018. [Online]. Available: http://energystorage.org. [Accessed 22 05 2018].

[38] C. G. Z., "Supercapacitor and supercapattery as emerging electrochemical energy stores," *International Materials Reviews,* vol. 62, no. 4, 2017.

[39] A. Colet-Subirachs, A. Ruiz-Alvarez, O. Gomis-Bellmunt, F. Alvarez-Cuevas-Figuerola and A. Sudria-Andreu, "Control of a utility connected microgrid," in *IEEE*, Gothenberg, Sweden, 2010.

[40] A. Elias-Alcega, M. Roman-Barri, A. Ruiz-Alvarez, I. Cairo-Molins, A. Sumper and O. Gomis-Bellmunt, "Implementation of a test microgrid in Barcelona," Frankfurt, 2011.

[41] A. Ruiz-Álvarez, A. Colet-Subirachs, O. Gomis-Bellmunt, M. Fernández-Mola, F. Álvarez-Cuevas-Figuerola, J. López-Mestre and A. Sudriá-Andreu, "Design, management and comissioning of a utility connected microgrid based on IEC 61850," in *Conference: Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, 2010.

[42] modbus.org, "MODBUS Application Protocol Specification V1.1b3," 2012.

[43] C. S., "Introduction to the Controller Area Network (CAN)," Texas Instruments, Austin, 2016.

[44] A. Ruiz-Álvarez, A. Colet-Subirachs, F. Álvarez-Cuevas Figuerola, O. Gomis-Bellmunt and A. Sudriá-Andreu, "Operation of a Utility Connected Microgrid Using an IEC 61850-Based Multi-Level Management System," *IEE Transactions on smart grid,* pp. vol. 3, no. 2., 2012.

[45] Scneider Electric, "Scneider Electric Website," 2018. [Online]. Available: http://www.wonderware.com/hmi-scada/intouch-machine-edition/. [Accessed 20 03 2018].

[46] M. Marzband, A. Sumper, A. Ruiz-Álvarez, J. Domínguez-García and B. and Tomoiaga, "Experimental evaluation of a real time energy management system for stand-alone microgrids in day-ahead markets," *Applied Energy,* pp. 106 pp. 365-376, 2013.

[47] S. A. Timoshin, "Control system with hysteresis and delay," *System & Control Letters,*

pp. 91 pp. 43-47, 2016.

[48] Shen, Z., Deng, W., Pei, W., Mu, L. and Ouyang, H., "Design and Implementation of Microgrid SCADA Platform," *Advanced Materials Research,* pp. 732-733, pp. 1358-1364., 2013.

[49] M. Marzband, A. Sumper, M. Chindris and B. Tomoiaga, "Energy Management System of Hybrid Microgrid with Energy Storage," *Buletinul AGIR,* p. no. 3, 2012.

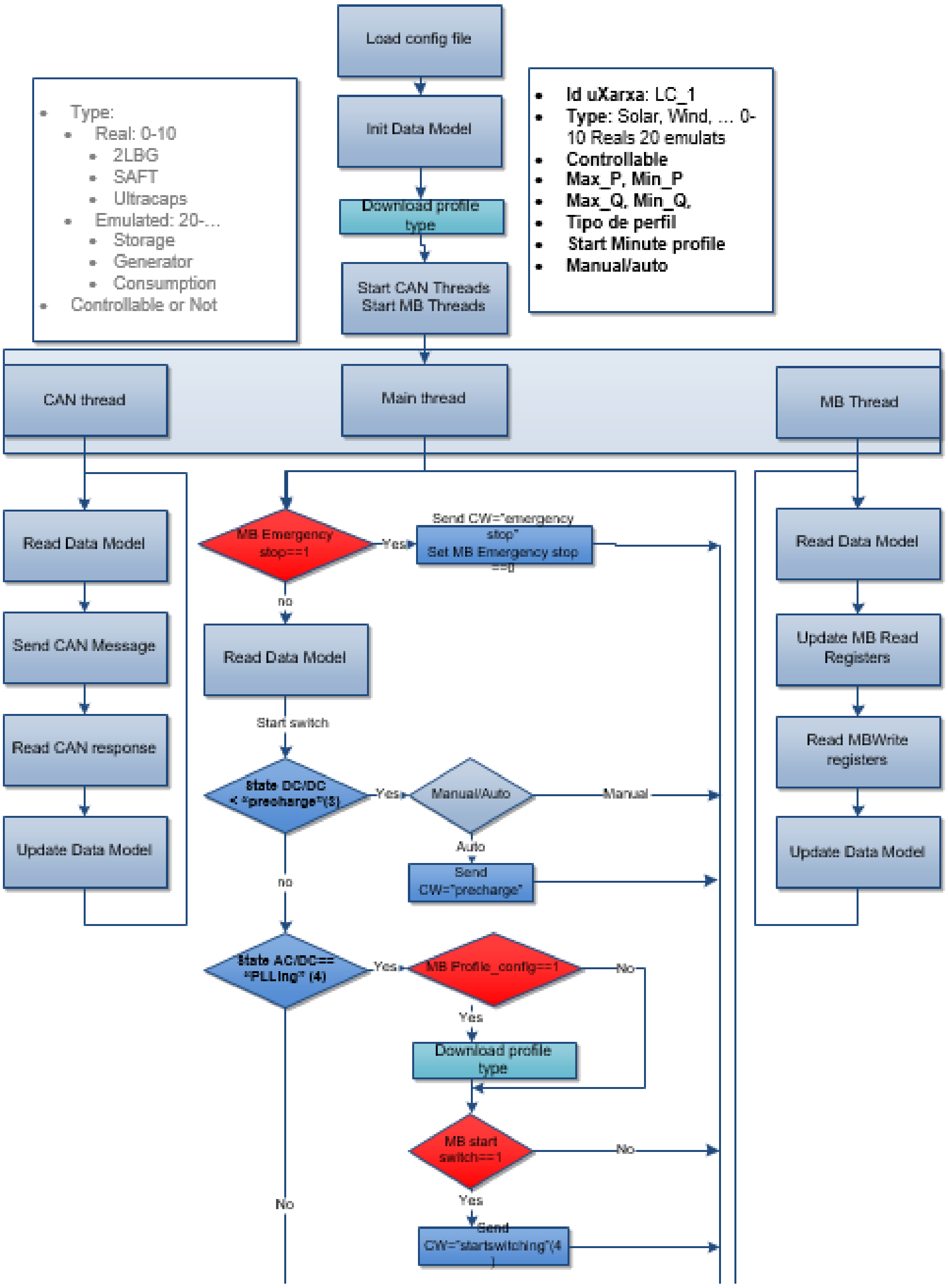
[50] G. Zapata, A. Salazar, M. D. and R. García, "Supervision of a Distributed Energy Resources Generation System Using IEC and ISA Standards," in *Robotics and Automation (CCRA)*, Bogota, Colombia, 2016.

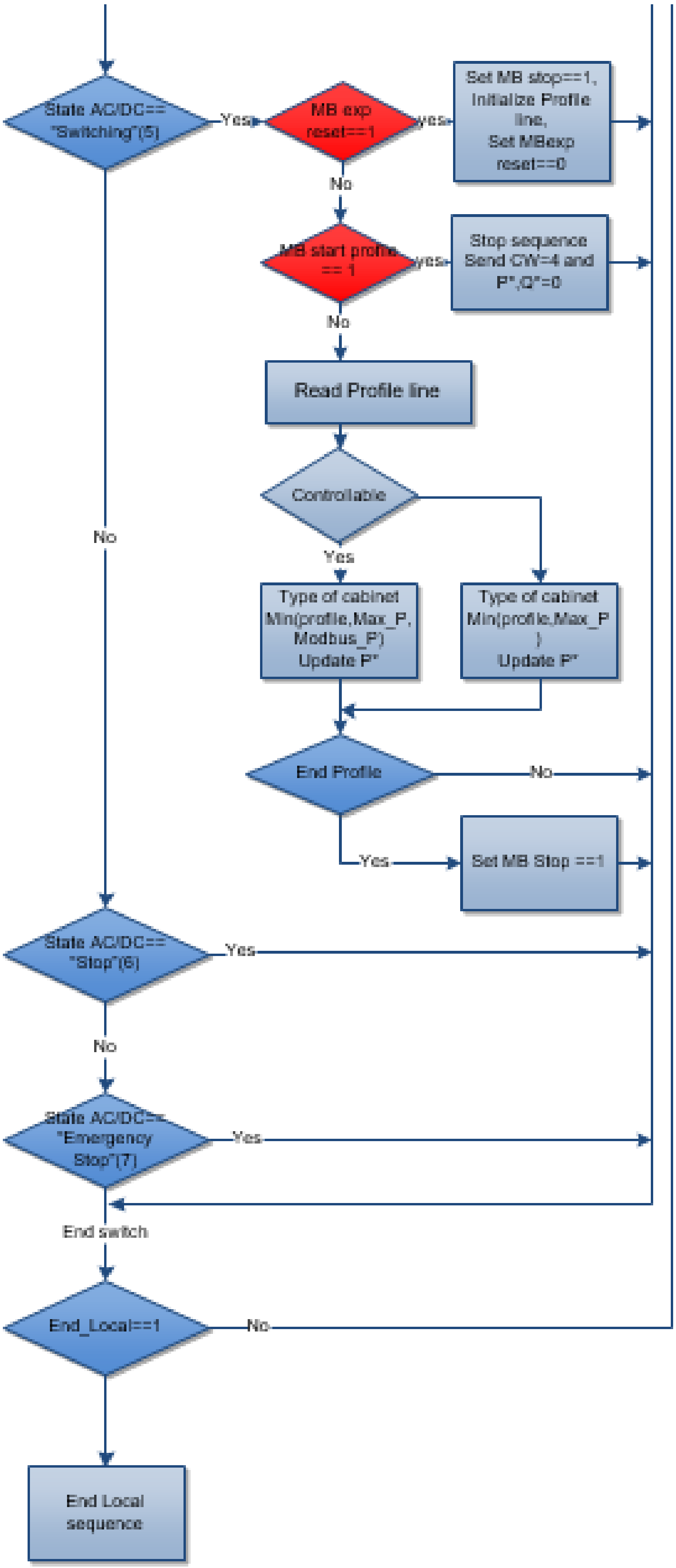
[51] IREC-ENDESA-CINERGIA, "Test Microgrid implemented in Barcelona (IREC) - Mechanisms for Multi-Microgrid environment management (ENDESA) - Power electronics and IEC61850 communications applied to a Microgrid system (CINERGIA)," Barcelona, 2011.

[52] M. I, D. L. Ferrer and C. J.-I., "Design and development of a test environment to analyze the impact of cyber attacks on the electrical distribution network," in *International Conference on Renewable Energies and Power Quality*, Madrid, 2016.

Appendix:

A. Flowchart of the LC state machine algorithm




State AC/DC== "Switching"(5)
Yes
MB exp reset==1
yes
Set MB stop==1, Initialize Profile line, Set MBexp reset==0
No
MB start profile == 1
yes
Stop sequence Send CW=4 and P*,Q*=0
No
Read Profile line
Controllable
Yes
Type of cabinet Min(profile,Max_P, Modbus_P) Update P*
Type of cabinet Min(profile,Max_P) Update P*
End Profile
No
Yes
Set MB Stop ==1
No
State AC/DC== "Stop"(6)
Yes
No
State AC/DC== "Emergency Stop"(7)
Yes
End switch
End_Local==1
No
End Local sequence

B. Detailed list of the laboratory equipments

Storage technology	Voltage	Power:	Energy:	Connection type:
Flywheel	400 V	5 kVA	30 kJ, 8.3 Wh	B2B converter
Lithium ion batteries	400V	5 kVA	6.9 MJ, 1.9 kWh	B2B converter
Ultracapacitors	400V	5 kVA	160 kJ, 44Wh	B2B converter
Storage emulations (up to 5 pieces)	400 V	5 kVA	∞	B2B converter with PLL in LSC

Power electronics:	Voltage:	Power:
Control of active and reactive power (in emulators, five units)	400 V	5 kVA (5)
Control of active and reactive power (in generation equipment, 4+1 equipment)	400 V	5 kVA (4) 30 kVA (1)

Generation technology:	Voltage:	Power:	Connection type:
DFIG (wind motor generator)	400 V	5 kVA	B2B converter
PMSG (wind motor generator)	400 V	5 kVA	B2B converter
SCIG (wind motor generator)	400 V	5 kVA	B2B converter
Emulator configured as generators	400 V	5 kVA	B2B converter with PLL in LSC
9-phased generator (wind motor generator)	400 V	30 kVA	Three B2B converters

Type of load:	Voltage:	Power:	Connection type:
Resistances	400 V	15 kW	Direct connection

Network signal generators:	Voltage:	Power:
Quality control of distribution network supply	Up to 600 V	200 kVA

C. Modbus mappings

Modbus address	Field	Reference in ITME	Unit	Functions 1500	Functions 1501	Type
Those highlighted in green are only available in the new configuration of the LCs						
0	Type of cabinet	RW_Type		r	rw	int16_t
1	Power Switch	R_PowerSwitch		r	r	int16_t
2	Maximum active power	RW_PMax	W	r	rw	int16_t
3	Minimum active power	RW_Pmin	W	r	rw	int16_t
4	Maximum reactive power	RW_QMax	var	r	rw	int16_t
5	Minimum reactive power	RW_QMin	var	r	rw	int16_t
6	Active power setpoint	RW_PSetPoint	W	rw	rw	int16_t
7	Reactive power setpoint	RW_QSetPoint	var	rw	rw	int16_t
10	State	RW_State		r	rw	int16_t
11	CAN State	R_State		r	r	int16_t
12	CAN Error	R_Error		r	r	int16_t
13	CAN read ID	RW_CANReadID		r	rw	int16_t
14	CAN transmit ID	RW_CANTransmitID		r	rw	int16_t
15	CAN mode	RW_ManAuto		r	rw	int16_t
16	CAN command	RW_CANCmd		r	rw	int16_t
17	CAN P	R_P	W	r	r	int16_t
18	CAN Q	R_Q	var	r	r	int16_t
19	CAN read 11 iterations	R_CANR11Iterations		r	r	uint16_t
20	CAN Iac	R_Iac	A	r	r	int16_t

21	CAN Vac	R_Vac	V	r	r	int16_t
22	CAN Frequency	R_Freq	Hz/100	r	r	int16_t
23	CAN read 12 iterations	R_CANR12Iterations		r	r	uint16_t
24	CAN Idc	R_Idc	A	r	r	int16_t
25	CAN Vdc	R_Vdc	V	r	r	int16_t
26	CAN Temperature	R_Temp	ºC	r	r	int16_t
27	CAN read 13 iterations	R_CANR13Iterations		r	r	uint16_t
28	CAN Type			r	rw	int16_t
29	CAN P setpoint			r	r	int16_t
30	CAN Q setpoint			r	r	int16_t
31 32	Modbus operation read iterations	R_MBRiterations		r	r	uint32_t
33 34	Modbus operation transmit iterations	R_MBTransmitIterations		r	r	uint32_t
35 36	Modbus operation read write iterations	R_MBRWIterations		r	r	uint32_t
40	File number power profile	RW_ProfileNum		r	rw	int16_t
41	Initial minute in power profile	RW_ProfileStartMin	m	r	rw	int16_t
42	Current minute in power profile	RW_ProfileCurrentMin	m	r	r	int16_t
43 44	Current iteration in power profile			r	r	int32_t

45	Type Set			r	r	int16_t
46	Samples per second			r	r	int16_t
47	Value of P in profile at current minute	RW_ProfileCur rentP	W	r	r	int16_t
48	EMS cab			r	rw	int16_t
51	SoC	R_SOC	%	r	r	int16_t
52	CAN State 2			r	r	int16_t
53	CAN RX 2			r	rw	int16_t
54	Initial iteration			r	rw	int16_t
55	Profile Config			r	rw	int16_t
56	Controllable	RW_Controllab le		r	rw	int16_t
57	start switch			r	rw	int16_t
58	start profile	RW_ProfileStar t		r	rw	int16_t
59	experiment reset			r	rw	int16_t
60	stop profile			r	rw	int16_t

D. Description of the key tags and classes

To introduce the developed SCADA system, it is crucial to give an introduction of the key tags and classes used. The following sections give an indication of the structure of the tags and classes created.

i.Tags created

Throughout the project it was the guideline was to solve the problems with as less tags as possible. It is important for keeping the SCADA's speed high, when later adding new features.

1. Tags used for the type-database

The following tags have been created for the purpose of storing the available types, used in availability conditions, scripts, and descriptions of the developed app. The structure has been built up in a way, so that new types can be easily added later, following the convention about real and emulated types.

1. **TypeDesc:** a string array defined in the startup script, aiming to be used for listing the available types and their modbus signal integer. In this array the type description is in the number of line corresponding to the integer code. As the type integer codes programmed in the locals are not in order, there are numerous lines being blank in this array.

One of InTouch Machine Edition's limitations found was numerous times not allowing inside-references of tags. For example, in the messages field of a list box, or when using a symbol on a project screen. This limitation has caused several problems, and the chosen way to solve it was to put the available type descriptions, emulated type descriptions, and their corresponding integer IDs in separated arrays.

2. **Types_Avail:** a string array defined in the startup script, being used to store the available type descriptions in the ascending order of their IDs.

3. **Types_Avail_ID:** an integer array defined in the startup script, being used to store the available type IDs in ascending order.

4. **Types_Avail_Emul:** a string array defined in the startup script, being used to store the available emulated type descriptions in the ascending order of their IDs.

5. **Types_Avail_Emul_ID**: an integer array defined in the startup script, being used to store the available emulated type IDs in ascending order.

The number of available types and emulated types are defined in **Types_AvailNo**, and in **Types_AvailEmulNo**.

2. Tags for configuration

The configuration of the writeable parameters of each LC are stored in a configuration .csv file. **Config_csv** indicates the path of this file. To make the structure of the configuration file treatment flexible, there has been an additional tag**, Config_csv_folder** created indicating the default folder. Moreover, the currently loaded file path is stored in **ConfigFilePath** and in **ConfigFilePath_Short**. The bite length of the top row of the configuration file is saved in

Config_Index.

3. Tags for profile-configurations

The profile files in .csv format are created in a folder named after the corresponding line of the Types_Avail tag. The string array **ProfConf_ProfileNames** contains the profile files of the selected folder, whose line number in Types_Avail is **ProfConf_SelectedFolder_No.** The selected file is always line number **ProfConf_SelectedFile_No** of the array containing the profile file names. **ProfConf_SelectedFolder_Path, ProfConf_SelectedFile_Path,** and **ProfConf_SelectedFile_ShortPath** are used for indicating the path of the selected folder and file. Following the same logic **ProfConf_SelectedFolder_Emul_No** is created for making references in the Types_Avail_Emul array. created. The length of the profile is given in the integer tag **ProfConf_DiagramLength.**

4. Script-related boolean tags

For running the scripts, there have been several booleans created. When any of these values are 1, the corresponding script runs. The correspondence is done according to the following table:

Name of the script:	Corresponding Boolean:
Config File Load	**Script_ConfigFileLoad**
Config File Save	**Script_ConfigFileSave**
Profile Configuration File Change	**Script_ProfcConf_FileChange**
Profile Configuration General Change	**Script_ProfConf_GeneralChange**
Modbus Update	**Script_UpdMB**

5. Miscellaneous

Several other tags have been created for the well-functioning of the SCADA. Throughout the program it is several times needed to select a LC, for example for more monitoring information. The currently selected LC is always given by the integer **Sel_LC**.

Other miscellaneous tags include **OpenedScreen**, **OpenedScreen_Sub**, **DiagramReload**, **DiagramStatus**, **DiagramStatus**, and **DBConnectionString**. OpenedScreen and OpenedScreen_Sub strings are used for indicating the opened screen, and opened sub screen of the program. **DiagramReload**, and **DiagramStatus** booleans are used for reloading a diagram on screen, and for showing the loading status. Lastly, the string **DBConnectionString**, defined in the startup script, is used for establishing the connection with the database.

6. Class tags

The array **cLC** corresponds to the class LocalController, and has a length of 9. Each line stands for a local controller in this tag.

For the configuration functions (load and save) there have been the class tag **Config_DataImport** created.

There has been an additional class created for the **demo** as well.

ii.Classes created

There have been two classes used throughout the project: **cLC**, and **Config_DataImport**.

1. cLC class

This class contains all the necessary data corresponding to one local controller, both from the SCADA, and from the local's level. All the class properties, along with their types and little explanations are listed below in the table. Properties starting with "R_", or "RW_" indicate "read" or "read-write", also indicating that these variables are linked to a modbus address. Property names starting without "R_" or "RW_" are usually used solely internally in the SCADA software.

cLC property	Type	Explanation
R_LCID	Integer	ID of the equipment
Connected	Boolean	1 Connected, 0 Not Connected
Enable	Boolean	1 YES, 0 NO
LCID_Desc	String	LCID Description under the figures (e.g. RB3)
LongDesc	String	Description line 3
MB_int16_ReadCompleted	Integer	For driver configuration
MB_int16_ReadStatus	Integer	For driver configuration
MB_int16_WriteCompleted	Integer	For driver configuration
MB_int16_WriteStatus	Integer	For driver configuration
MB_int32_ReadCompleted	Integer	For driver configuration

MB_int32_ReadStatus	Integer	For driver configuration
MB_int32_WriteCompleted	Integer	For driver configuration
MB_int32_WriteStatus	Integer	For driver configuration
Name	String	Description line 1
ProfileNum	Integer	File number power profile On SCADA
R_CANR11Iterations	Integer	No. of CAN messages received by package 11
R_CANR12Iterations	Integer	No. of CAN messages received by package 12
R_CANR13Iterations	Integer	No. of CAN messages received by package 13
R_Error	Integer	Error code
R_Freq	Real	Frequency
R_Iac	Real	AC current
R_Idc	Real	DC Current
R_MBRiterations	Real	Modbus operation read iterations
R_MBRWIterations	Real	Modbus operation read write iterations
R_MBTransmitIterations	Real	Modbus operation transmit iterations
R_P	Real	Active Power Injected
R_PowerSwitch	Boolean	Power Switch
R_Q	Real	Reactive Power Injected
R_SOC	Real	SoC
R_State	Integer	State of the converters 2.
R_Temp	Real	Temperature
R_Vac	Integer	AC Voltage
R_Vdc	Real	DC Voltage
RW_CANCmd	Integer	CAN Command
RW_CANReadID	Integer	CAN identificator for reading messages
RW_CANTransmitID	Integer	CAN identificator for sending messages
RW_Controllable	Boolean	1 YES, 0 NO
RW_ManAuto	Integer	Manual 20, Auto 10
RW_PMax	Real	Max. Active Power
RW_Pmin	Real	Min. Active Power
RW_ProfileActualized	Integer	1 update, 0 nothing, -1 updating
RW_ProfileCurrentMin	Real	Current Minute in Power Profile
RW_ProfileCurrentP	Real	Value of P in current minute
RW_ProfileNum	Integer	File Number Power Profile
RW_ProfileStart	Real	Start reading file
RW_ProfileStartMin	Real	Initial Minute in Power Profile
RW_PSetPoint	Real	Active Power Setpint
RW_QMax	Real	Max. Reactive Power
RW_QMin	Real	Min. Reactive Power
RW_QSetPoint	Real	Reactive Power Setpoint

RW_Start	Boolean	1 start, 0 nothing
RW_State	Integer	State of the converters
RW_Stop	Boolean	1 stop, 0 nothing
RW_Type	Integer	Type of the equipment, MB (1 - Ultracap, 2 - Real Battery, - 20 - Res., 21 - PV, 22 - Wind, 23 - Battery, 24 - Grid)
RW_TypeDesc	String	MB Read Type Description
ShortDesc	String	Description line 2
StationID	String	Example: 192.168.0.101:1500:1
Type	Integer	Type of the equipment, SCADA
TypeDesc	String	SCADA Write Type Description

2. Config_DataImport class

The Config_DataImport class contains as many lines, as many properties are stored in the .csv configuration file. Following the nomenclature of the configuration file, the following table shows the properties of this class, including which Data peace is linked to which property imported or exported.

Data_1	Integer	Vector Index
Data_2	Integer	R_LCID
Data_3	String	LCID_Desc
Data_4	Boolean	Enabled
Data_5	String	StationID
Data_6	String	Name
Data_7	String	ShortDesc
Data_8	String	LongDesc
Data_9	Integer	Type
Data_10	Integer	RW_ManAuto
Data_11	Integer	RW_Controllable
Data_12	Real	RW_Pmax
Data_13	Real	RW_Pmin
Data_14	Real	RW_Qmax
Data_15	Real	RW_Qmin
Data_16	Integer	ProfileNum

3. Demo class

Class property	Type
CaseToRun	Integer
InitialValues	Boolean
K	Integer

Pd	Real
Pd_lim	Real
Pgrid	Real
Pgrid_setpoint	Real
ProfileStartMin	Integer
Psaftmax	Real
RemainingTime	Integer
RunningTime	Integer
Runtime	Integer
SOC_max	Integer
SOC_min	Integer
Start_TimeStamp	Integer
State	Integer
Tdelay	Integer
Z	Integer

Z_delay has been created as an integer for being able to follow its timestamp.

E. Table of error codes of the locals (R_Error) and their meaning

Code, R_Error:	Meaning:
1	Drivers
2	Overcurrent
3	Drivers and overcurrent
4	Bus overvoltage
5	Driver and bus overcurrent dc
6	Overload of bus dc and overcurrent
7	Drivers and bus overcurrent
8	Overheating
9	Overtemperature and drivers
10	Overheating and overcurrent
11	Overtemperature, drivers and overcurrent
12	Overtemperature and overvoltage bus
13	Overtemperature, drivers and bus overvoltage
14	Overtemperature, overvoltage and overcurrent
15	Overtemperature, drivers, bus overvoltage and overcurrent
20	Frequency error
21	PLL error
22	Network voltage error, V<Vbrexa
23	Network voltage error, V>Vrbxa

ETSEIB

24	Preloading error
25	Ambient temperature error
26	Inconsistent message, COD
27	Incosistent message, P
28	Inconsistent message, Q
29	Incompatible message, CW
40	Emergency stop made by iSocket
41	Emergency stop made by the other converter
60	Error in the CAN bus
61	Time out for CAN bus
62	Communication error between converters

F. Developed functions

i.Update the description of the types

```
Option Explicit

' SCRIPT FOR UPDATING TYPE DESCRIPTIONS (E.G. SOLAR)
Function UpdTypeDesc()

Dim i
i=0
Do While i<=$cLC->Size
$cLC[i].TypeDesc=$TypeDesc[$cLC[i].Type]
$cLC[i].RW_TypeDesc=$TypeDesc[$cLC[i].RW_Type]

i=i+1
Loop

End Function
```

ii.Update ProfConf_ProfileNames[]

```
'$region: Configuration of type and profilenum
'SCRIPT FOR UPDATING PROFCONF_PROFILENAMES ARRAY WITH THE
CURRENT FOLDER'S PROFILE NAMES

Function UpdateProfileNames()

'/ 1. Erasing current elements from profilenames
Dim i, k
i=0
Do While i<=$ProfConf_ProfileNames->Size
        $ProfConf_ProfileNames[i]=k
```

```
'         $ProfConf_ProfileNums
          i=i+1
Loop
'/ END of 1. Erasing current elements from profilenames

'/ 2. Filling up the ProfConf_ProfileNames array with the current folder's profile names
$ProfConf_SelectedFolderFileNo=$FindFile($ProfConf_SelectedFolder_Path&"*.csv","ProfConf_ProfileNames",1000)
'/ END of 2. Filling up the ProfConf_ProfileNames array with the current folder's profile names

End Function
```

G. Developed codes

i. Startup script

```
'Variables available for all Script groups from the Script task can be declared and initialized here.
'Procedures available for all Script groups from the Script task can be implemented here.


'------------------OWN SCRIPT STARTS-------------
'Functionalities:
'         1.Definition of important variables (filenames)
'         2.Config file information load in tags
'         3.Database config
'         4.Type definitions
'   5. Update Modbus variables

'-------------------------------------------------------------------------


'--------------------'         1.Definition of important variables (filenames)--------------------------------------------------

Dim Config_Filename, Config_header
Config_FileName="IREC_UGRID_CONFIG" 'Name of Config file under Config\LC_Config_Files\
Config_header=$GetAppPath() & "Config\LC_Config_Files\" & Config_FileName&"_HEADER.csv"

$Config_csv=$GetAppPath() & "Config\LC_Config_Files\" & Config_FileName&".csv"
$Config_csv_folder=$GetAppPath() & "Config\LC_Config_Files\"
$Config_Index=$FileReadFields(Config_header,0,"Config_DataImport.Data_1",16)

$ConfigFilePath=$Config_csv_folder&$ConfigFilePath_ShortSave

$OpenedScreen="Configurations"
```

```
'--------------------END of 1.Definition of important variables (filenames)----------------------------
------------------------



'---------------2.Config file information load in tags---------------
Dim i, Index, k, l
i=0
Index=$Config_Index 'Skipping the first line of the config file

'Default values
i=0
Do While i<=$cLC->Size
	$cLC[k].RW_ProfileNum=0
	$cLC[k].RW_Type=0
	$cLC[k].RW_TypeDesc=0
	$cLC[k].R_PowerSwitch=0
	$cLC[k].RW_PMax=0
	$cLC[k].RW_Pmin=0
	$cLC[k].RW_QMax=0
	$cLC[k].RW_QMin=0
	$cLC[k].RW_PSetPoint=0
	$cLC[k].RW_QSetPoint=0
	$cLC[k].RW_State=0
	$cLC[k].R_State=0
	$cLC[k].R_Error=0
	$cLC[k].RW_CANReadID=0
	$cLC[k].RW_CANTransmitID =0
	$cLC[k].RW_CANCmd=0
	$cLC[k].R_P=0
	$cLC[k].R_Q=0
	$cLC[k].R_CANR11Iterations=0
	$cLC[k].R_Iac =0
	$cLC[k].R_Vac=0
	$cLC[k].R_Freq=0
	$cLC[k].R_CANR12Iterations=0
	$cLC[k].R_Idc=0
	$cLC[k].R_Vdc=0
	$cLC[k].R_Temp=0
	$cLC[k].R_CANR13Iterations=0
	$cLC[k].R_MBRiterations=0
	$cLC[k].R_MBTransmitIterations=0
	$cLC[k].R_MBRWIterations=0
	$cLC[k].RW_ProfileNum=0
	$cLC[k].RW_ProfileStartMin=0
	$cLC[k].RW_ProfileCurrentMin=0
	$cLC[k].RW_ProfileStart=0
	$cLC[k].RW_ProfileCurrentP=0
```

```
        $cLC[k].R_SOC=0
        $cLC[k].RW_Controllable=0
        $cLC[k].RW_Start=0
        $cLC[k].RW_Stop=0
        $cLC[k].RW_ProfileActualized=0


                i=i+1
Loop

i=0
Do While i<=$cLC->Size

        Index=$FileReadFields($Config_csv,Index,"Config_DataImport.Data_1",16)

                $cLC[i].R_LCID=$Config_DataImport.Data_2
                $cLC[i].LCID_Desc=$Config_DataImport.Data_3
                $cLC[i].Enable=$Config_DataImport.Data_4
                $cLC[i].StationID=$Config_DataImport.Data_5
                $cLC[i].Name=$Config_DataImport.Data_6
                $cLC[i].ShortDesc=$Config_DataImport.Data_7
                $cLC[i].LongDesc=$Config_DataImport.Data_8
                $cLC[i].Type=$Config_DataImport.Data_9
                $cLC[i].RW_ManAuto=$Config_DataImport.Data_10
                $cLC[i].RW_Controllable=$Config_DataImport.Data_11
                $cLC[i].RW_Pmax=$Config_DataImport.Data_12
                $cLC[i].RW_Pmin=$Config_DataImport.Data_13
                $cLC[i].RW_Qmax=$Config_DataImport.Data_14
                $cLC[i].RW_Qmin=$Config_DataImport.Data_15
                $cLC[i].ProfileNum=$Config_DataImport.Data_16

                i=i+1
Loop
'---------------END of 2.Config file information load in tags---------------

'---------------3. Database config---------------

$DBConnectionString="Provider=MYSQLCLIENT; Server=127.0.0.1; Database=scada; Uid=root; Pwd=;"

'--------------------END of  3. Database config----------------------------------------------------

'---------------4. Type definitions---------------
$TypeDesc[0]="-1" 'Real ultracap
$TypeDesc[1]="Ucap" 'Real ultracap
$TypeDesc[2]="Re_Battery" 'Real battery
$TypeDesc[20]="Resid" 'Emulation
$TypeDesc[21]="Solar" 'Emulation
$TypeDesc[22]="Wind" 'Emulation
$TypeDesc[23]="Em_Battery" 'Emulated battery
$TypeDesc[24]="Grid" 'Emulation
```

```
Call UpdTypeDesc()

'Listing the available type codes
i=0 'For the general loop
l=1 'For the vector of types available - starting from 1 (because of later functions'
characteristics
k=1 'For the vector of emulated types available - starting from 1 (because of later
functions' characteristics
Do While i<=$TypeDesc->Size
        If $StrLen($TypeDesc[i])>2 Then
                $Types_Avail_ID[l]=$TypeDesc[i]->Index
                $Types_Avail[l]=$TypeDesc[i]

                'Getting the emulated types
                If ($Types_Avail_ID[l]>19 And $Types_Avail[l]<>"Grid") Then 'Over 19
there are the emulated types, "grid" cannot be chosen
                        $Types_Avail_Emul_ID[k]=$TypeDesc[i]->Index
                        $Types_Avail_Emul[k]=$TypeDesc[i]
                        $Types_AvailEmulNo=k 'No of available emulated types
                        k=k+1
                End If

                $Types_AvailNo=l 'No of available types
                l=l+1
End If


i=i+1
Loop


'--------------------END of  4. Type definitions-----------------------------------------------------

'---------------5. Update Modbus variables---------------

i=0
Do While i<=$cLC->Size
        $cLC[i].MB_int16_ReadStatus=-1
        $cLC[i].MB_int16_WriteStatus=-1
        $cLC[i].MB_int32_ReadStatus=-1
        $cLC[i].MB_int32_WriteStatus=-1
        i=i+1
Loop

$Script_UpdMB=1
```

```
'-------------------END of  5. Updating Modbus variables-----------------------------------------------
-----
```

ii. Modbus updating script

```
'Variables available only for this group can be declared here.
'The code configured here is executed while the condition configured in the Execution field
is TRUE.


'-----------------OWN SCRIPT STARTS-------------
'Functionalities:
'	1.Checking all corresponding drivers, changing connected status
'	2.Update MB variables (currently disabled for the old configs)


'Check if connected
Dim i
i=0
Do While i<=$cLC->Size
If $cLC[i].MB_int16_ReadStatus=0  And $cLC[i].MB_int16_Conf_ReadStatus=0 Then
'later also put cLC[i].MB_int32_Conf_ReadStatus, if it is used
	$cLC[i].Connected=1
Else
	$cLC[i].Connected=0
End If
	i=i+1
Loop

'What to do if connected, what to do if not.
'Dim k
'k=0
'Do While k<=$cLC->Size
'If $cLC[k].Enable=1 Then
'	$cLC[k].RW_ProfileNum=$cLC[k].ProfileNum 'this shall run from the SCADA state
machine in case of the new locals.
'	$cLC[k].RW_ProfileStartMin=$Demo.ProfileStartMin
'	$cLC[k].RW_Type=$cLC[k].Type


'	If k<>3 Then
'	$cLC[k].RW_CANReadID=3
'	$cLC[k].RW_CANTransmitID =2
'	Else
'	$cLC[k].RW_CANReadID=1
'	$cLC[k].RW_CANTransmitID =2
'	End If
'End If
'k=k+1
'Loop

$Script_UpdMB=0
```

iii. Configuration file load

```
'Variables available only for this group can be declared here.
'The code configured here is executed while the condition configured in the Execution field
is TRUE.


'------------------OWN SCRIPT STARTS-------------
'Comments:
'	The script should run after clicking and choosing a config file
'Functionalities:
'1.Load chosen config file on the tags
'-----------------------------------------------------------------------------

$FileCopy($ConfigFilePath,$Config_csv,1000)
'---------------1.Load chosen config file on the tags -> update current config file---------------

Dim csv,i, Index
Index=$Config_Index 'Skipping the first line of the config file


i=0
Do While i<=$cLC->Size

	Index=$FileReadFields($ConfigFilePath,Index,"Config_DataImport.Data_1",16)

		'Vector ID (currently the same as LCID) is not loaded, as not used in
SCADA
		$cLC[i].R_LCID=$Config_DataImport.Data_2
		$cLC[i].LCID_Desc=$Config_DataImport.Data_3
		$cLC[i].Enable=$Config_DataImport.Data_4
		$cLC[i].StationID=$Config_DataImport.Data_5
		$cLC[i].Name=$Config_DataImport.Data_6
		$cLC[i].ShortDesc=$Config_DataImport.Data_7
		$cLC[i].LongDesc=$Config_DataImport.Data_8
		$cLC[i].Type=$Config_DataImport.Data_9
		$cLC[i].RW_ManAuto=$Config_DataImport.Data_10
		$cLC[i].RW_Controllable=$Config_DataImport.Data_11
		$cLC[i].RW_Pmax=$Config_DataImport.Data_12
		$cLC[i].RW_Pmin=$Config_DataImport.Data_13
		$cLC[i].RW_Qmax=$Config_DataImport.Data_14
		$cLC[i].RW_Qmin=$Config_DataImport.Data_15
		$cLC[i].ProfileNum=$Config_DataImport.Data_16

		i=i+1
Loop

Call UpdTypeDesc()
```

```
'---------------END of 1.Load chosen config file on the tags--------------
$Script_ConfigFileLoad=0
```

iv. Configuration file save

```
'Variables available only for this group can be declared here.
'Variables available only for this group can be declared here.
'The code configured here is executed while the condition configured in the Execution field
is TRUE.


'------------------OWN SCRIPT STARTS-------------
'Comments:
'	The script should run after clicking and choosing a file directory and name
'Functionalities:
'	1.Update config file
'	2.Save current config file on the selected folder and name

'---------------	1.Update-------------------------

'---------------	1.Update config file with the current tags-------------------------
Dim Index,i,Config_update, Config_header, Config_FileName

Config_update=$GetAppPath() & "Config\LC_Config_Files\" &
Config_FileName&"_update.csv"
Config_FileName="IREC_UGRID_CONFIG" 'Name of Config file under
Config\LC_Config_Files\
Config_header=$GetAppPath() & "Config\LC_Config_Files\" &
Config_FileName&"_HEADER.csv"


$FileCopy(Config_header,Config_update, 1000) 'Header file replaces the auxiliary
updating config file

Index=$Config_Index 'Skipping the first line of the config file
i=0 'Starting from the 0th LC
Do While i<=$cLC->Size

		$Config_DataImport.Data_1=$cLC[i].R_LCID 'Vector ID
		$Config_DataImport.Data_2=$cLC[i].R_LCID
		$Config_DataImport.Data_3=$cLC[i].LCID_Desc
		$Config_DataImport.Data_4=$cLC[i].Enable
		$Config_DataImport.Data_5=$cLC[i].StationID
		$Config_DataImport.Data_6=$cLC[i].Name
		$Config_DataImport.Data_7=$cLC[i].ShortDesc
		$Config_DataImport.Data_8=$cLC[i].LongDesc
		$Config_DataImport.Data_9=$cLC[i].Type
		$Config_DataImport.Data_10=$cLC[i].RW_ManAuto
		$Config_DataImport.Data_11=$cLC[i].RW_Controllable
		$Config_DataImport.Data_12=$cLC[i].RW_Pmax
		$Config_DataImport.Data_13=$cLC[i].RW_Pmin
		$Config_DataImport.Data_14=$cLC[i].RW_Qmax
```

```
                $Config_DataImport.Data_15=$cLC[i].RW_Qmin
                $Config_DataImport.Data_16=$cLC[i].ProfileNum


        Index=$FileWriteFields(Config_update,Index,"Config_DataImport.Data_1",16)

                i=i+1
Loop

$FileCopy(Config_update,$Config_csv, 1000) 'overwrite config file

$FileDelete(Config_update) 'delete auxiliary update file
'---------------     END of 1.Update config file with the current tags-------------------------


'---------------     2.Save current config file on Saved_ConfigFiles folder, based on the
chosen -------------------------
$FileCopy($Config_csv,$ConfigFilePath, 1000) 'Copy original file to the "old" directory
'---------------     END of 2.Save current config file on Saved_ConfigFiles folder, based on
date-------------------------


$Script_ConfigFileSave=0

'The code configured here is executed while the condition configured in the Execution field
is TRUE.
```

v. Profile configuration general change

```
'Variables available only for this group can be declared here.

'/ 1. Only if the opened screen is Configurations.SCC
        If $OpenedScreen="Configurations" And $OpenedScreen_Sub="" Then
        '/ 1.1. -----------Internal variable definitions

                Dim typeid, profilenum, profstring, i, id, lcid
                lcid=$Sel_LC
                typeid=$cLC[lcid].Type 'Type ID
                profilenum=$cLC[lcid].ProfileNum 'ProfileNum integer
                profstring=$Str(profilenum)&".csv" 'ProfileNum string


        '/ 1.2. -----------Searching for type folder path -  ProfConf_SelectedFolder_Path

        $ProfConf_SelectedFolder_Path=$GetAppPath()&"Database\Profiles\"&$TypeDes
c[typeid]&"\" 'selected folder
                        Call UpdateProfileNames() ' ----------Updating array with
filenames
                        $ProfConf_SelectedFolder_Emul_No=1
```

```
        '/ 1.3. -----------Searching for ProfConf_SelectedFile_No
                        i=1
                Do While i<=$ProfConf_ProfileNames->Size
                        If $ProfConf_ProfileNames[i]=profstring Then
                                $ProfConf_SelectedFile_No=$ProfConf_ProfileNames[i]-
>Index

        $ProfConf_SelectedFile_Path=$ProfConf_SelectedFolder_Path&$ProfConf_Profil
eNames[i]
                        End If
                        i=i+1
                Loop
        End If


'/ END of 1. Only if the opened screen is Configurations.SCC

'/ 2. Only if the opened screen is Profile_Configuration.SCC
        If $OpenedScreen="Profile_Configuration" Then

        $ProfConf_SelectedFolder_Path=$GetAppPath()&"Database\Profiles\"&$Types_A
vail[$ProfConf_SelectedFolder_No]&"\" 'selected folder
                $ProfConf_SelectedFile_No=1
                Call UpdateProfileNames() ' ----------Updating array with filenames
        End If
'/ END of 2. Only if the opened screen is Profile_Configuration.SCC

'/ 3. Only if the opened screen is Configurations.SCC and the opened subscreen is
Configurations_TypeChange.SCC
        If $OpenedScreen_Sub="Configurations_TypeChange" And
$OpenedScreen="Configurations" Then

        $cLC[$Sel_LC].Type=$Types_Avail_Emul_ID[$ProfConf_SelectedFolder_Emul_N
o]

        $ProfConf_SelectedFolder_Path=$GetAppPath()&"Database\Profiles\"&$TypeDes
c[$cLC[$Sel_LC].Type]&"\" 'selected folder path
                Call UpdateProfileNames() ' ----------Updating array with filenames
                Call UpdTypeDesc() 'Updating TypeDescs
                $cLC[$Sel_LC].ProfileNum=$StrGetElement($ProfConf_ProfileNames[1],
".", 1)
        End If
'/ 3. Only if the opened screen is Configurations.SCC and the opened subscreen is
Configurations_TypeChange.SCC

'/ 4. ----------Running file-changing script
$Script_ProfConf_FileChange=1
'/ END of 4. ----------Running file-changing script
```

```
$Script_ProfConf_GeneralChange=0

'The code configured here is executed while the condition configured in the Execution field
is TRUE.
```

vi. Profile configuration file change

```
'Variables available only for this group can be declared here.

'Functionalities:
'	1.Check path of the file, defining $ProfConf_SelectedFile_Path and
	$ProfConf_SelectedFile_ShortPath
'	2.Calculating the length of file selected

'/ 1. Check path of the file, defining $ProfConf_SelectedFile_Path and
$ProfConf_SelectedFile_ShortPath
	$ProfConf_SelectedFile_Path=$ProfConf_SelectedFolder_Path&$ProfConf_Profil
eNames[$ProfConf_SelectedFile_No] 'For lookupload it is needed (diagramlength)
	$ProfConf_SelectedFile_ShortPath="Database\Profiles\"&$Types_Avail[$ProfConf
_SelectedFolder_No]&"\"&$ProfConf_ProfileNames[$ProfConf_SelectedFile_No] 'For
opening the excel file it is needed
'/ END of 1. Check path of the file, defining $ProfConf_SelectedFile_Path and
$ProfConf_SelectedFile_ShortPath

'/ 2. Calculating the length of file selected
$ProfConf_DiagramLength=$LookupLoad($ProfConf_SelectedFile_Path,1,1,",")
'/ END of 2. Calculating the length of file selected

$Script_ProfConf_FileChange=0

'The code configured here is executed while the condition configured in the Execution field
is TRUE.
```

vii. Demo screen States

```
'Variables available only for this group can be declared here.

'Parameters: Demo.Runtime + parameters of SCADAControl defined on the Demo
Screen.

Dim i, TimeNow


If $Demo.State=-1 Then '0s
	i=0
	Do While i<=$cLC->Size
		If $cLC[i].Enable=1 Then 'only Demo-participant LCs are enabled.
```

```
                                        $cLC[6].RW_Type=10 'At the beginning of the test we have
to send a Type=0
                                        $cLC[4].RW_Type=15 'At the beginning of the test we have
to send a Type=0
                        End If
                        i=i+1
            Loop
End If

If $Demo.State=0 Then 'PREPARE
            'Updating in order 1. ProfleNum, 2. ProfileStartMin, 3. Type
            Dim k
            k=0
            Do While k<=$cLC->Size
            If $cLC[k].Enable=1 Then
                        $cLC[k].RW_ProfileNum=$cLC[k].ProfileNum 'this shall run from the
SCADA state machine in case of the new locals.
                        $cLC[k].RW_ProfileStartMin=$Demo.ProfileStartMin
                        $cLC[k].RW_Type=$cLC[k].Type

                        If k<>3 Then
                        $cLC[k].RW_CANReadID=3
                        $cLC[k].RW_CANTransmitID =2
                        Else
                        $cLC[k].RW_CANReadID=1
                        $cLC[k].RW_CANTransmitID =2
                        End If
            End If
            k=k+1
            Loop
            End If

If $Demo.State=1 Then 'STARTING
            i=0
            Do While i<=$cLC->Size
                        If $cLC[i].Enable=1 Then
                                    $cLC[i].RW_ManAuto=10
                        End If
            i=i+1
            Loop
            $Demo.InitialValues=1 'InitialValuesFor the start of Control
            $Script_SCADAControl=1 'Control on
            $Demo.Start_TimeStamp=$DateTime2Clock($Date,$Time) 'Stamping Current
Time in sec from 1970
            $Demo.State=2

End If

If $Demo.State=2 Then 'RUNNING
            TimeNow=$DateTime2Clock($Date,$Time) 'Current Time in sec from 1970
                        If TimeNow-$Demo.Start_TimeStamp>$Demo.Runtime*60 Then
```

```
                $Demo.State=3
                End If
End If

If $Demo.State=3 Then 'STOP
        $Script_SCADAControl=0 'Control off
        $Demo.Start_TimeStamp=0
        i=0
        Do While i<=$cLC->Size
                If $cLC[i].Enable=1 Then
                        $cLC[i].RW_ManAuto=20
                End If
        i=i+1
        Loop
        $Demo.State=-1
        $Demo.Runningtime=0
        $Demo.RemainingTime=0
End If

$Demo.RunningTime=TimeNow-$Demo.Start_TimeStamp
$Demo.RemainingTime=$Demo.Runtime*60-$Demo.RunningTime

'The code configured here is executed while the condition configured in the Execution field
is TRUE.
```

viii. SCADA state machine

```
'Variables available only for this group can be declared here.

Dim TimeOutProfile
TimeOutProfile=120

Dim k
k=0

'In the planned state machine only state=4 (PLLing) implies programming tasks
Do While k<=$cLC->Size
If $cLC[k].Enable=1 And $cLC[k].R_State=4 Then 'PLLing
        If $cLC[k].ProfileNum<>$cLC[k].RW_ProfileNum Then 'If profile number is not
actualized
                If $cLC[k].RW_ProfileActualized=0 Then
                        $cLC[k].RW_ProfileActualized=1
                        $cLC[k].ProfileLastUpdate=$DateTime2Clock($Date,$Time) ' Time
in sec from 1970
                Else
                If $cLC[k].ProfileLastUpdate>TimeOutProfile Then
                        'SHOWERROR CODE
                End If
                End If
```

```
        End If
        Else
                'ENABLE BUTTON TO START
End If
k=k+1
Loop
```

'The code configured here is executed while the condition configured in the Execution field is TRUE.

ix. Control code

```
'Variables available only for this group can be declared here.

'PARAMETERS:
'Definition of Demo.CaseToRun, Demo.Pd_lim, Demo.Psaftmax, Demo.SOC_max,
Demo.SOC_min, Demo.Tdelay on the demo screen

'Definition of Pd demand
$Demo.Pd=$cLC[6].R_P+$cLC[4].R_P

'DEFINITION OF THE BASIC CONTROL/HYSTERESIS CURVE
'Case 1 Conditions - Basic Control
If $Demo.CaseToRun=1 Then
$Demo.K=1
End If

'Case 2 Conditions - Control Based on Hysteresis
If $Demo.CaseToRun=2 Then

        'Initial definition of Z
        If $Demo.InitialValues=1 Then
                If $Demo.Pd>0 Then
                        $Demo.Z=-1
                        $Demo.InitialValues=0
                Else
                        If $Demo.Pd<0 Then
                                $Demo.Z=1
                                $Demo.InitialValues=0
                        Else
                                $Demo.Z=0
                                $Demo_Zdelay=0
                                $Demo.K=0
                        End If
                End If
        End If

        'Definition of Zdelay
        If $Demo.Pd<-$Demo.Pd_lim Or $Demo.Pd>$Demo.Pd_lim Then
                If $Demo.Pd>$Demo.Pd_lim Then
                        $Demo_Zdelay=-1
                ElseIf $Demo.Pd<-$Demo.Pd_lim Then
```

```
                $Demo_Zdelay=1
            End If
        End If

        'Calculation of Z depending on Zdelay
        Dim TimeNow, TimeZChange
        TimeZChange=$DateTime2Clock(($StrGetElement($Demo_Zdelay->TimeStamp,"
",1)),($StrGetElement($Demo_Zdelay->TimeStamp," ",2))) 'TimeStamp in sec from 1970
        TimeNow=$DateTime2Clock($Date,$Time) 'Current Time in sec from 1970
        If  TimeNow-TimeZChange>$Demo.Tdelay Then
            $Demo.Z=$Demo_Zdelay
        End If

        'Calculation of K
        If $Demo.Z=-1 Then
            If  $Demo.Pd>0 Then
                $Demo.K=1
            Else
                $Demo.K=0
            End If
        End If

        If $Demo.Z=1 Then
            If  $Demo.Pd<0 Then
                $Demo.K=1
            Else
                $Demo.K=0
            End If
        End If
End If
'End of Case 2 Conditions - Control Based on Hysteresis
'END OF THE DEFINITION OF THE BASIC CONTROL/HYSTERESIS CURVE


'COMMON PART OF CASE 1-2 STARTS
If $Demo.Pd<=0 Then 'Overproduction
        If $cLC[3].R_SOC<$Demo.SOC_max Then
            $cLC[3].RW_PSetPoint=-$Demo.K*$Max($Demo.Pd,-$Demo.Psaftmax) '-
because of the sign convenctions
        Else
            $cLC[3].RW_PSetPoint=0
        End If
Else 'Extra power needed
        If $cLC[3].R_SOC<=$Demo.SOC_min Then
            $cLC[3].RW_PSetPoint=0
        Else
            $cLC[3].RW_PSetPoint=-$Demo.K*$Min($Demo.Pd,$Demo.Psaftmax) '-
because of the sign convenctions
```

```
        End If
End If
'End of COMMON PART OF CASE 1-2

$Demo.Pgrid_SetPoint=$cLC[3].RW_PSetPoint+$cLC[4].R_P+$cLC[6].R_P
$Demo.Pgrid=$cLC[3].R_P+$cLC[4].R_P+$cLC[6].R_P

'The code configured here is executed while the condition configured in the Execution field
is TRUE.
```

H. Description of the SCADA screens

i.Configuration screen

Screen style: overlapped

Save and load button

Save button:
Command on down:
$RDFileN("$ConfigFilePath",$GetAppPath()&"\Configurations\SavedFiles\", "*.csv", 1)
Command on up: $Script_ConfigFileSave=1

Load button:
Command on down:
$RDFileN("$ConfigFilePath",$GetAppPath()&"\Configurations\SavedFiles\", "*.csv", 1)
Command on up:$Script_ConfigFileLoad=1

Configuration symbol (Configuration_LC.sym)

Used inside variables, and their reference properties in the LocalController class

Inner variable of the symbol:	Reference example, LC 9:
Main.Connected	cLC[9].Connected
Main.DescriptionLine1	cLC[9].Name
Main.DescriptionLine2	cLC[9].ShortDesc
Main.Enabled	cLC[9].Enable
Main.LCID	cLC[9].R_LCID
Main.LCID_Desc	cLC[9].LCID_Desc
Main.ProfileNum	cLC[9].ProfileNum
Main.R_Error	cLC[9].R_Error
Main.R_SoC	cLC[9].R_SoC
Main.RW_Controllable	cLC[9].RW_Controllable
Main.RW_ManAuto	cLC[9].RW_ManAuto
Main.RW_Pmax	cLC[9].RW_Pmax
Main.RW_Pmin	cLC[9].RW_Pmin
Main.RW_ProfileNum	cLC[9].RW_ProfileNum
Main.RW_Qmax	cLC[9].RW_Qmax
Main.RW_Qmin	cLC[9].RW_Qmin

Main.RW_Type	cLC[9].RW_Type
Main.Type	cLC[9].Type
Main.TypeDesc	cLC[9].TypeDesc

Equipment symbol:

Visibility of each equipment symbol, example:
#LCID:=4

Type indicator:

Visibility of each type symbol, example:
#TypeDesc:="Em_Battery" OR #TypeDesc:="Re_Battery"

Profile updating symbol:

Visibility:
#Enabled:=1 AND #Type:>19

Command on down:
$OpenedScreen_Sub=""
$Sel_LC=#LCID:
$Script_ProfConf_GeneralChange=1

Command on up:
$Close("Configurations_ProfileChange")
$Open("Configurations_ProfileChange")

Type updating symbol

Visibility:
#Type:>19 AND #TypeDesc:<>"Grid" AND #Enabled:=1

Command on down::
$Sel_LC=#LCID:
$Script_ProfConf_GeneralChange=1

Command on up:
$Close("Configurations_TypeChange")
$Open("Configurations_TypeChange")

Visibilities of the backgrounds:

Visibility of the purple background:
#Type:>19 AND #Enabled:=1

Visibility of the red background:
#Type:>19 AND #Enabled:=1

Visibility of the blue background:
#Type:<10 AND #Enabled:=1

Visibility of the orange frame:
#LCID:=Sel_LC AND (OpenedScreen_Sub="Configurations_ProfileChange" OR OpenedScreen_Sub="Configurations_TypeChange")

Visibility of the disconnected X and the connecting line:

Visibility of the red X:
#Connected:<>1 AND #Enabled:=1

Visibility of the red line behind:
(#Connected:=1 OR #Enabled:=0) AND #R_Error:=0

Visibilities and commands in the selector pane of manual/auto and control/no control:

Command on down, control manual/auto rectangle:
#RW_ManAuto:=10 OR #RW_ManAuto:=20

Command on down, control control rectangle:
#RW_Controllable:=0 OR #RW_Controllable:=1

Visibilities of control manual/auto rectangle:
#RW_ManAuto:=20 OR #RW_ManAuto:=10

Visibilities of control manual/auto rectangle:
#RW_Controllable:=1 OR #RW_Controllable:=0

The color animations used are not detailed here.

ii.Configuration profile change screen

Screen style: dialog
Screen logic:
On open: OpenedScreen_Sub="Configurations_ProfileChange"

Screen scripts
While open:

```
If $ProfConf_SelectedFile_No>$ProfConf_SelectedFolderFileNo Then
$ProfConf_SelectedFile_No=$ProfConf_SelectedFolderFileNo
End If


If $OpenedScreen_Sub<>"Configurations_ProfileChange" Then
        $Close("Configurations_ProfileChange")
End If
```

Profile Selector list box:
Object properties:
User enable: 1
Control enable: 1
Read/Search Tag: ProfConf_SelectedFile_No
Write Tag: ProfConf_SelectedFile_No
Messages and values (until 5)

Message:	Value:
No profile	0
{ProfConf_ProfileNames[1]}	1
{ProfConf_ProfileNames[2]}	2
{ProfConf_ProfileNames[3]}	3
{ProfConf_ProfileNames[4]}	4
{ProfConf_ProfileNames[5]}	5

Diagram
Points
Llabel: Profile{ProfConf_SelectedFile_Path}
Data source: csv
Tag field: 1
***Data sources*:**
Reload: DiagramReload
Data source settings: {ProfConf_SelectedFile_Path

OK button:
Command on down:
```
$OpenedScreen="Configurations"
```

Command on up:
```
Dim profileoriginal, profileid

If $ProfConf_SelectedFile_No<>0 Then
profileoriginal=$ProfConf_ProfileNames[$ProfConf_SelectedFile_No]
profileid =$NCopy(profileoriginal,0, $StrLen(profileoriginal)-4) 'skipping .csv
Else
profileid=0
End If

$cLC[$Sel_LC].ProfileNum=profileid

$Close("Configurations_ProfileChange")
$Open("Configurations")
```

Cancel button:
Command on up:
```
$Close("Configurations_ProfileChange")
$Open("Configurations")
```

iii.Configuration type change screen

Screen style: dialog
Screen logic:
```
On open: OpenedScreen_Sub= "Configurations_TypeChange"
On close:
OpenedScreen_Sub=""
Sel_LC=99
```

Screen scripts
On open:
```
$Script_ProfConf_GeneralChange=1
```
While open:
```
If $ProfConf_SelectedFolder_Emul_No>$Types_AvailEmulNo Then
$ProfConf_SelectedFolder_Emul_No=$Types_AvailEmulNo
End If

If $OpenedScreen_Sub<>"Configurations_TypeChange" Then
        $Close("Configurations_TypeChange")
End If
```

Type Selector list box:
Object properties:
User enable: 1
Control enable: 1
Read/Search Tag: ProfConf_SelectedFolder_Emul_No
Write Tag: ProfConf_SelectedFolder_Emul_No

Messages and values (until 5)

Message:	Value:
{Types_Avail_Emul[1]}	1
{Types_Avail_Emul[2]}	2
{Types_Avail_Emul[3]}	3
{Types_Avail_Emul[4]}	4
{Types_Avail_Emul[5]}	5

OK button

OK button:
Command on down:
$Script_ProfConf_GeneralChange=1
Command on up:
$Open("Configurations")
$Close("Configurations_TypeChange")
Cancel button
Command on up:
$Close("Configurations_TypeChange")
$Open("Configurations")

iv.Profile configuration screen

Screen scripts
While open:

```
If $ProfConf_SelectedFile_No>$ProfConf_SelectedFolderFileNo Then
$ProfConf_SelectedFile_No=$ProfConf_SelectedFolderFileNo
End If

If $ProfConf_SelectedFolder_No>$Types_AvailNo Then
$ProfConf_SelectedFolder_No=$Types_AvailNo

End If
```

Type selector list box
Object properties:
User enable: 1
Control enable: 1
Read/Search Tag: ProfConf_SelectedFolder_No
Write Tag: ProfConf_SelectedFolder_No
Messages and values (until 5)

Message:	Value:
{Types_Avail[1]}	1
{Types_Avail[2]}	2
{Types_Avail[3]}	3
{Types_Avail[4]}	4
{Types_Avail[5]}	5

File selector list box
Object properties:
User enable: 1
Control enable: 1
Read/Search Tag: ProfConf_SelectedFile_No

Write Tag: ProfConf_SelectedFile_No

Messages and values (until 5)

Message:	Value:
{ProfConf_ProfileNames[1]}	1
{ProfConf_ProfileNames[2]}	2
{ProfConf_ProfileNames[3]}	3
{ProfConf_ProfileNames[4]}	4
{ProfConf_ProfileNames[5]}	5

Modify profile in Excel Button

Command on down:

```
Dim programtorun
programtorun="C:\Program Files (x86)\Microsoft Office\Office15\EXCEL.exe"
$WinExec(programtorun&" "&$ProfConf_SelectedFile_ShortPath)
```

Diagram

Points

Llabel: Profile{ProfConf_SelectedFile_Path}
Data source: csv
Tag field: 1

***Data sources*:**

Reload: DiagramReload
Data source settings: {ProfConf_SelectedFile_Path

v.Monitoring screen

Monitorization symbol (Monitorization_LC.sym)

Used inside variables, and their reference properties in the LocalController class

Inner variable of the symbol:	Reference example, LC 9:
Main.Connected	cLC[9].Connected
Main.DescriptionLine1	cLC[9].Name
Main.DescriptionLine2	cLC[9].ShortDesc
Main.Enabled	cLC[9].Enable
Main.LCID	cLC[9].R_LCID
Main.LCID_Desc	cLC[9].LCID_Desc
Main.R_Error	cLC[9].R_Error
Main.R_Iac	cLC[9].R_Iac
Main.R_Idc	cLC[9].R_Idc
Main.R_P	cLC[9].R_P
Main.R_Q	cLC[9].R_Q
Main.R_SoC	cLC[9].R_SoC
Main.R_State	cLC[9].R_State
Main.R_Vac	cLC[9].R_VaC
Main.R_Vdc	cLC[9].R_Vdc

Main.RW_CANCmd	cLC[9].RW_CANCmd
Main.RW_ManAuto	cLC[9].RW_ManAuto
Main.RW_PSetPoint	cLC[9].RW_PSetPoint
Main.RW_QSetPoint	cLC[9].RW_QSetPoint
Main.RW_Start	cLC[9].RW_Start
Main.RW_Stop	cLC[9].RW_Stop
Main.RW_TypeDesc	cLC[9].RW_TypeDesc
Main.Type	cLC[9].Type

Equipment symbol:
Visibility of each equipment symbol, example:
#LCID:=4

Type indicator:
Visibility of each type symbol, example:
#TypeDesc:="Em_Battery" OR #TypeDesc:="Re_Battery"

Visibilities of the backgrounds:
Visibility of the purple background:
#Type:>19 AND #Enabled:=1

Visibility of the red background:
#Type:>19 AND #Enabled:=1

Visibility of the blue background:
#Type:<10 AND #Enabled:=1

Visibility of the orange frame:
#LCID:=Sel_LC AND (OpenedScreen_Sub="Configurations_ProfileChange" OR OpenedScreen_Sub="Configurations_TypeChange")

Visibility of the disconnected X and the connecting line:
Visibility of the red X:
#Connected:<>1 AND #Enabled:=1

Visibility of the red line behind:
(#Connected:=1 OR #Enabled:=0) AND #R_Error:=0

SW monitoring
Visibility of the green rectangle:
#R_State:=1

CW monitoring and control
Object properties of manual mode, start precharging radio button:
Caption: Start prech.
Tag: #RW_CANCmd:
True value: 3
Disable: #R_State:>3

Object properties of manual mode, start radio button:
Caption: Start
Tag: #RW_CANCmd:
True value: 4
Disable: #R_Vdc:<700 AND #R_State:=4

Object properties of manual mode, stop radio button:
Caption: Stop
Tag: #RW_CANCmd:
True value: 5

Visibility of all manual mode, all radio buttons:
#RW_ManAuto:=20

Visibility of auto mode, “Waiting for auto precharge” text:
#RW_ManAuto:=10 AND #R_State:<4

Visibility of auto mode, start button:
#RW_ManAuto:=10 AND #R_State:=4 AND #R_Vdc:>700

Command on down of auto mode, start button:
#RW_Start:=1

Visibility of auto mode, stop button:
#RW_ManAuto:=10 AND #R_State:>=4 AND #R_State:<=6

Command on down of auto mode, stop button:
#RW_Stop:=1

Visibility of the “Emergency stop” button:
#R_State:<>7

Command on down of the “Emergency stop” button:
#RW_CANCmd:=1

Command on down of the “Click for more” button:
$Sel_LC=#LCID:

Command on up of the “Click for more” button:
$Close("Monitoring_More")

vi.Detailed monitoring screen

Screen style: dialog
Title bar: Detailed monitoring of LC {Sel_LC}
Screen logic:
On open: OpenedScreen_Sub= "Monitoring_More"
On close:
OpenedScreen_Sub=""
Sel_LC 99

Symbol of More_LC:

More_LC symbol; used inside variables, and their reference properties in the LocalController class

Inner variable of the symbol:	Reference example, CAN id. to read msg.
Main.Data	cLC[Sel_LC].RW_CANReadID
Main.Description	CAN id. to read msg.
Main.Unit	cLC[Sel_LC].RW_CANReadID->Unit

vii.Alarm screen

Alarm object with no filter.

viii.History screen

Trend control:

Data sources:

Data Source: DB
Source Type: Database
X axis field: Time_Stamp
Max. Buffer: 1024
Database connection string: {DBConnectionString}
Table name: automatically create, trends_p

Points:

Label:	Data Source:	Tag/Field
LC3_R_P	DB	LC3_R_P
LC3_RW_PSetPoint	DB	LC3_RW_PSetPoint
LC4_R_P	DB	LC4_R_P
LC4_RW_PSetPoint	DB	LC4_RW_PSetPoint
LC6_R_P	DB	LC6_R_P
LC6_RW_PSetPoint	DB	LC6_RW_PSetPoint

ix.TNAVBAR bar screen

Tab button symbol (tabbutton.sym)

Used inside variables, and their example value in case of the button for Configurations

Inner variable of the symbol:	Reference example, CAN id. To reas msg.
Main.Label	Configurations
Main.ScreenToOpen	Configurations

Orange button:
Visibility:
$Sel_LC=#LCID
$OpenedScreen=#ScreenToOpen:

Command on down:
$OpenScreen(“#ScreenToOpen:”)

Command on up:
$OpenedScreen=#ScreenToOpen:
$Sel_LC=99

Green button:
Visibility:
$Sel_LC=#LCID
$OpenedScreen<>#ScreenToOpen:

Command on down:
$OpenScreen("#ScreenToOpen:")

Command on up:
$OpenedScreen=#ScreenToOpen:
$Sel_LC=99

I. Matlab source code of the simulation

```
clear all

%Importing data

load Phase_4_InputData_CloudCut
time=cell2mat(Input(:,1)) % s
Pcons=cell2mat(Input(:,2)) % W
Ppv=cell2mat(Input(:,3)) % W

%Input parameters
   %Fixed parameters
E_initial=5.04*10^7 %J; %input data read from RB16 screen
E_max=7.2*10^7 %J; 10 kWh %data got from the battery poster descr
SOC_min=20; % % %data from Andreas Sumpers article
SOC_max=80; % % %data from Andreas Sumpers article
P_saftmax=4000; %W %limitation of the DC/DC converter (rated power of the battery is 150000W

   %Control Changeable parameters
ControlCases=[2]; %Case 1 - basic; Case 2 - hysteresis
ControlCases_Pd_lim=[300]; %W
ControlCases_Tdelay=[120]; %second
NoSc=length(ControlCases) %Number of Scenarios which would run

%matrix creation of the important calculating parameters
Z=zeros(length(Ppv),NoSc) % -
Zdelay=zeros(length(Ppv),NoSc) % -
K=zeros(length(Ppv),NoSc) % -
SOC=zeros(length(Ppv),NoSc) % %
E_battery=zeros(length(Ppv),NoSc) % J
Zchangestamp=zeros(length(Ppv),NoSc) % -  % time-timestamp of Z
E_grid=zeros(length(Ppv),NoSc) % J
Pgrid=zeros(length(Ppv),NoSc) % W
Psaft=zeros(length(Ppv),NoSc) % W
Pd=zeros(length(Ppv),NoSc) % W
```

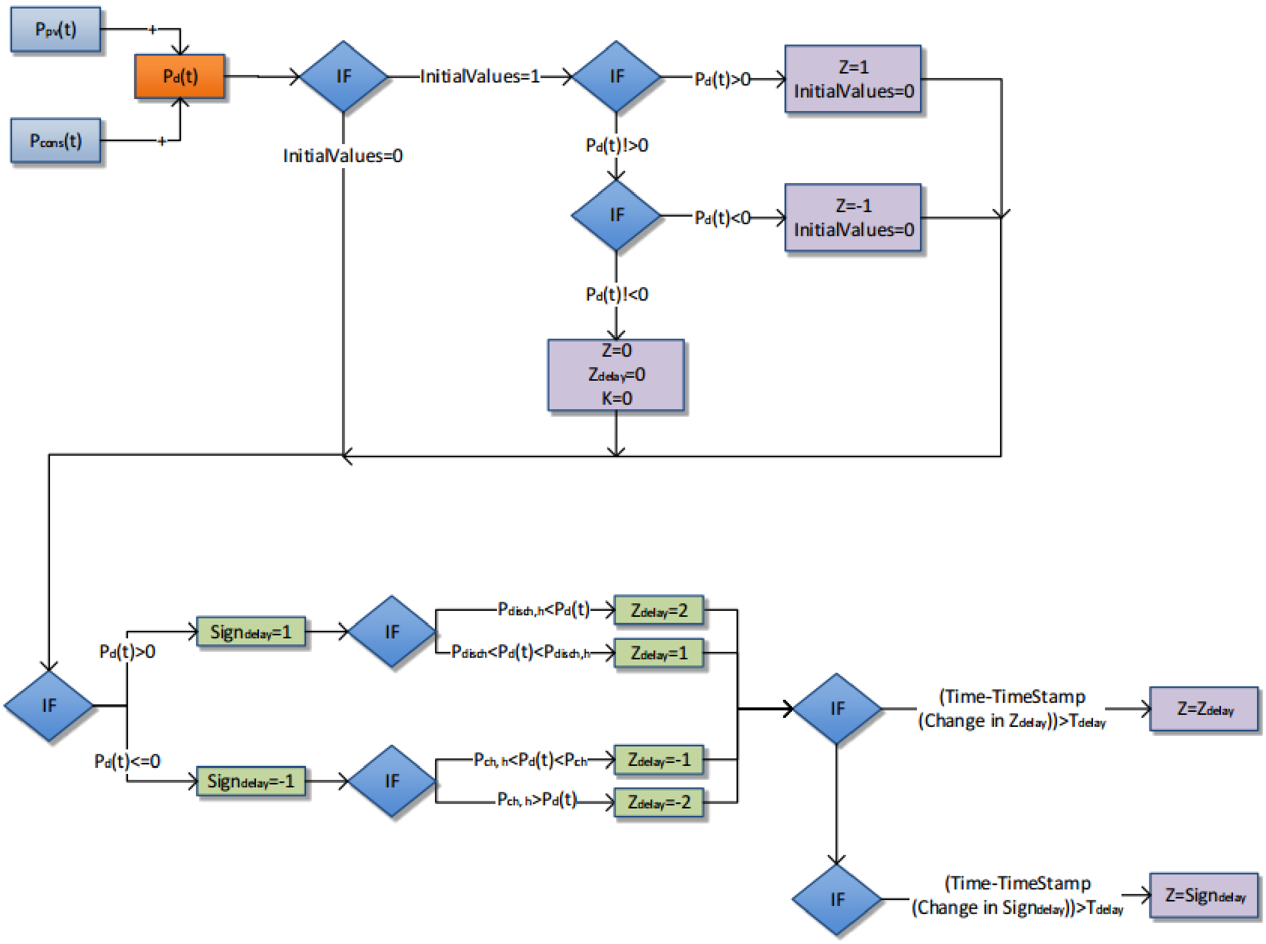
```
%Calculation starts
for sc=1:NoSc %NoSc

%Corresponding control parameters
CaseToRun=ControlCases(sc)
Pd_lim=ControlCases_Pd_lim(sc)
Tdelay=ControlCases_Tdelay(sc)

%First second calculations
E_battery(1,sc)=E_initial % W
SOC(1,sc)=E_initial*100/E_max

Control_Start=1 % starting condition for the start of control
i=1; %starting second
while i<=7201
   %basic calculations
   Pd(i,sc)=Pcons(i,1)+Ppv(i,1)

   if i~=1
   SOC(i,sc)=E_battery(i-1,sc)*100/E_max
   end


   %Control case differentiation
   if CaseToRun==1
      K(i,sc)=1
   end
   if CaseToRun==2
      %Initial Z calculation
      if Control_Start==1
         if Pd(i,sc)>0
            Z(i,sc)=-1
            Zdelay(i,sc)=-1
         else
            Z(i,sc)=1
            Zdelay(i,sc)=1
         end
         Control_Start=0
         Zchangestamp(1,sc)=0
      end

      %Calculation of Zdelay
      if i~=1
         if Pd(i,sc)<-Pd_lim || Pd(i,sc)>Pd_lim %checking if Pd is out of the limits
            if Pd(i,sc)>Pd_lim %checking if Pd is out of the positive limits
               Zdelay(i,sc)=-1
               if Zdelay(i-1,sc)~=-1 %putting timestamp if it has changed
                  Zchangestamp(i,sc)=0
               else % if it has not changed then increase Zchangestamp by one
                  Zchangestamp(i,sc)=Zchangestamp(i-1,sc)+1
               end
            elseif Pd(i,sc)<-Pd_lim %checking if Pd is out of the negative limits
               Zdelay(i,sc)=1
               if Zdelay(i-1,sc)~=1 %putting timestamp if it has changed
                  Zchangestamp(i,sc)=0
```

```
            else % if it has not changed then increase timestamp by one
              Zchangestamp(i,sc)=Zchangestamp(i-1,sc)+1 % if it has not changed then increase
Zchangestamp by one
            end
          end
        else
          Zdelay(i,sc)=Zdelay(i-1,sc) % if it is not out of any limits then Z stays the same
          Zchangestamp(i,sc)=Zchangestamp(i-1,sc)+1    % if it is not out of any limits then increase
Zchangestamp by one
        end
      end

     %Calculation of Z depending on Zdelay
     if i~=1
        if Zchangestamp(i,sc)>=Tdelay %change Z when Zchangestamp is over Tdelay
          Z(i,sc)=Zdelay(i,sc)
        else
          Z(i,sc)=Z(i-1,sc)
        end
     end
     %Calculation of K
      if Z(i,sc)==-1
        if Pd(i,sc)>0
          K(i,sc)=1
        else
          K(i,sc)=0
        end
      elseif Z(i,sc)==1
        if Pd(i,sc)<0
          K(i,sc)=1
        else
          K(i,sc)=0
        end
      end
   end
   %End of Control Case 2

   %Common part of Case 1-2 starts
   if Pd(i,sc)<=0 %overproduction
        if SOC(i,sc)<SOC_max
          Psaft(i,sc)=K(i,sc)*max(Pd(i,sc),-P_saftmax)
        else
          Psaft(i,sc)=0
        end
   end
   if Pd(i,sc)>0 %extra power needed
        if SOC(i,sc)>SOC_min
          Psaft(i,sc)=K(i,sc)*min(Pd(i,sc),P_saftmax)
        else
          Psaft(i,sc)=0
        end
   end
```

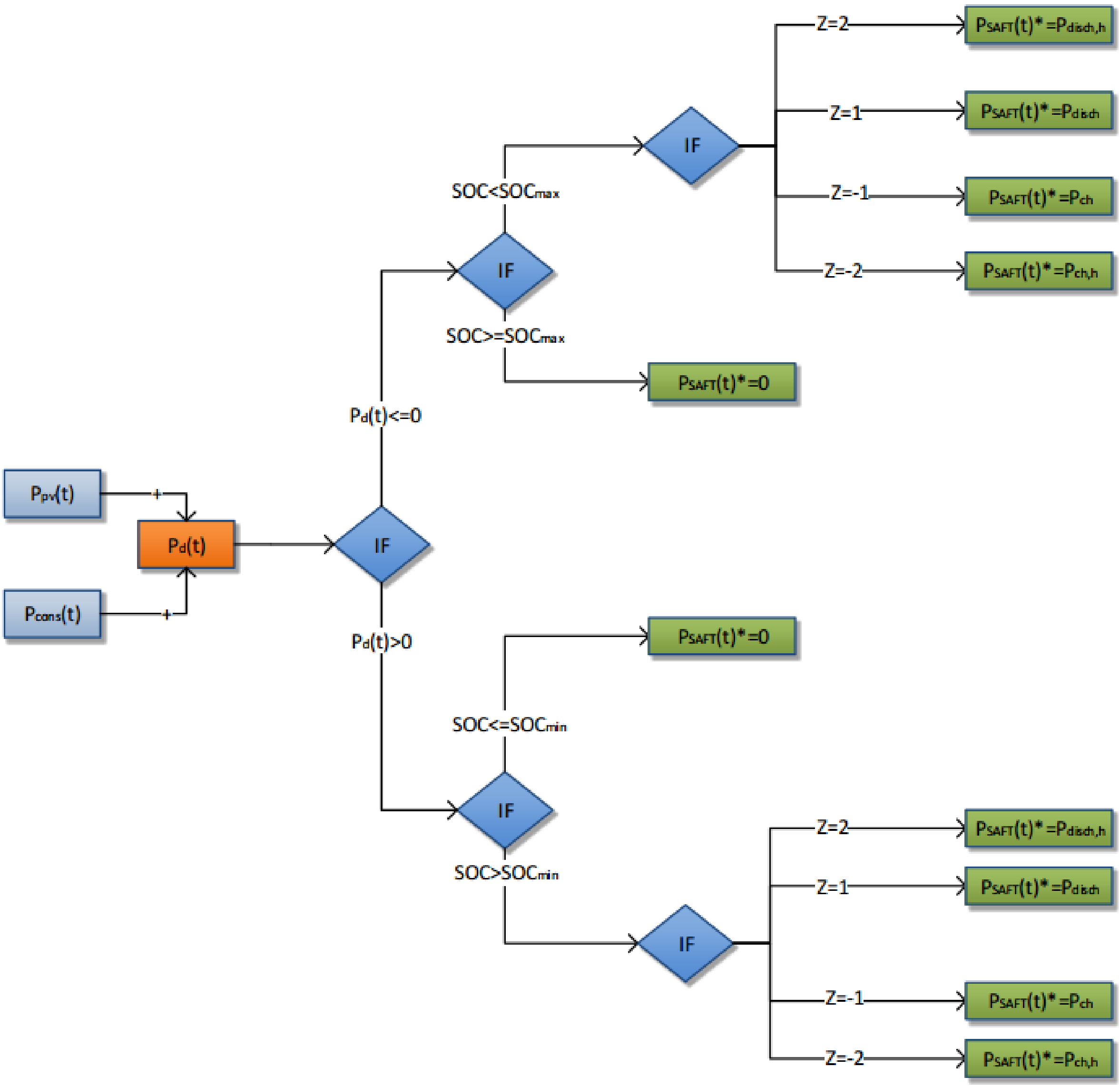
```
    Pgrid(i,sc)=Pd(i,sc)-Psaft(i,sc)
    E_grid(i,sc)=sum(Pgrid(:,sc))
    E_battery(i,sc)=E_battery(1,sc)-sum(Psaft(:,sc)) %minus because of the power convention | +: discharging battery -: charging battery

  i=i+1

end %end of a calculation of a second
end %end of a Control Case
```

J. Additional Control Idea, flowchart

This control uses 4 pre-defined P_{saft}* values. A normal charging (P_{ch}), a high charging ($P_{ch,\ h}$), a normal discharging (P_{disch}) and a high discharging ($P_{disch.\ h}$) control values, each corresponding to a pre-defined P_{limit} values. When the value of P_d is over or below a specific limit over the delay time, the response is the change of P_{saft}* to one of these pre-defined values.



ETSEIB



Ppv(t)
Pcons(t)
+
+
Pd(t)
IF
Pd(t)<=0
Pd(t)>0
IF
SOC<SOCmax
SOC>=SOCmax
IF
Z=2
Z=1
Z=-1
Z=-2
PSAFT(t)*=Pdisch,h
PSAFT(t)*=Pdisch
PSAFT(t)*=Pch
PSAFT(t)*=Pch,h
PSAFT(t)*=0
PSAFT(t)*=0
SOC<=SOCmin
IF
SOC>SOCmin
IF
Z=2
Z=1
Z=-1
Z=-2
PSAFT(t)*=Pdisch,h
PSAFT(t)*=Pdisch
PSAFT(t)*=Pch
PSAFT(t)*=Pch,h

K. Emulation event logging of 5th June, 2018

Time: 5th June 2018 – 16:20 – 18:20

Events:

- The emulation has started at 16:20. All cabinets started switching normally. The data save was continuous.
- At 16:49 (12:29 of the emulated day) emulation cabinet no. 4 (with LC RB04, PV emulation) switched to emergency mode because of an internal sequence of the power converter and stopped operating. The time was noted, the equipment was restarted and by 16:56 (12:36 of the emulated day) the equipment continued switching.
- At 17:20 (13:00 of the emulated day) emulation cabinet no. 4 (with LC RB04, PV emulation) switched to emergency mode because of an internal sequence of the power converter and stopped operating. The cabinet was restarted by 17:30 (13:10 of the emulated day) from the first point of power-generation, causing a shift in the profile.
- At 18:20 the demo has stopped all cabinets.

Decision:

Because of the issues logged above, the emulation on the 5th June was decided to be repeated.